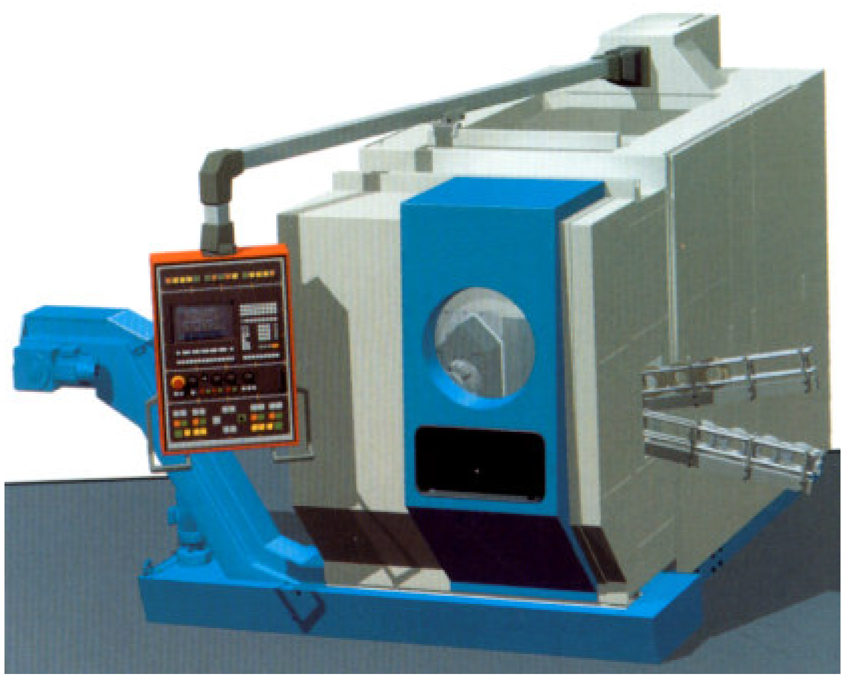
Grzegorz Nikiel
Akademia Techniczno-Humanistyczna
w Bielsku-Białej
Katedra Technologii Maszyn i Automatyzacji

Programowanie obrabiarek CNC na przykładzie układu sterowania Sinumerik 810D/840D



Bielsko-Biała 2004

Spis treści

WSTĘP

Znajomość zagadnienia tworzenia programów sterujących dla obrabiarek CNC staje się coraz bardziej pożądaną umiejętnością. Zrozumienie funkcjonowania układów CNC i zasad programowania nie jest szczególnie trudne. Niestety, brak na polskim rynku wydawniczym w miarę aktualnych, dostosowanych do potrzeb dydaktyki publikacji na ten temat. Trudno również o łatwo dostępne programy symulacyjne, pozwalające na praktyczną naukę programowania.

Bazując na doświadczeniach wyniesionych z prowadzenia zajęć na Akademii Techniczno-Humanistycznej (dawniej Filii Politechniki Łódzkiej), jak również kursów programowania i obsługi układów sterowania, w szczególności firmy Siemens, opracowano niniejszy skrypt. Jest on poświęcony programowaniu obrabiarek CNC w oparciu o popularny język sterowania Sinumerik 810D/840D (f. Siemens). Obecnie to jeden z najbardziej rozbudowanych układów sterowania stosowanych w Polsce, posiadający ogromne możliwości, szeroko stosowany zarówno w małych jak i dużych firmach. Jest on doskonałą podstawą do poznawania języków innych układów sterowania, podobnie jak Sinumerik najczęściej bazujących na tzw. G-kodach.

W skrypcie omawiane są podstawowe zasady funkcjonowania układów sterowania CNC, przede wszystkim wynikające z zależności pomiędzy układami współrzędnych (rozdz. 1). Przedstawiono najważniejsze z układów współrzędnych, jakie wymagane są do poprawnej pracy obrabiarki, sterowanej z poziomu programu sterującego. W syntetyczny sposób zaprezentowano zagadnienia przygotowania obrabiarki do pracy w trybie automatycznym. Znajomość zagadnień z tego rozdziału nie jest wymagana dla wprowadzenia do podstaw programowania (rozdz. 2), tym niemniej na dalszych etapach nauki, jak i przy nauczaniu podstaw obsługi obrabiarek CNC okaże się na pewno niezbędna.

W rozdz. 2 w przystępny sposób omówiono najważniejsze elementy programu sterującego: blok, słowo, adres. Przedstawiono podstawowe adresy języka Sinumerik 810D/840D. Bardzo szczegółowo potraktowano programowanie ruchów narzędzia, w tym podstawowe rodzaje interpolacji (rozdz. 3), transformacje układów współrzędnych (rozdz. 4). Omówiono zagadnienie kompensacji promienia narzędzia dla obróbki tokarskiej i frezarskiej (rozdz. 5). Dużo miejsca poświęcono bardziej zaawansowanym technikom programowania, takim jak programowanie parametryczne (rozdz. 8), podprogramy (rozdz. 9) i związanych z nimi cyklom obróbkowym jako podstawowej metodzie automatyzacji tworzenia programów sterujących (rozdz. 10). Zaprezentowano także rozwiązywanie specyficznych zagadnień, takich jak programowanie obróbki gwintów (rozdz. 6). Przedstawiono także praktyczne aspekty przygotowania programów sterujących na tle całego procesu planowania wytwarzania (rozdz. 11).

Wiele informacji zostało zilustrowanych prostymi przykładami programów obróbczych. Zostały one tak przygotowane, by w pierwszej fazie można było metodą symulacji sprawdzić przebieg programu i instrukcji w nim zawartych, a następnie próbować samodzielnych zmian i modyfikacji, na bieżąco weryfikując ich skutki. Do symulacji programów sterujących zaleca się stosować program autora niniejszego skryptu – ProgMaster. Jest to przykład systemu CAM do komputerowego wspomagania projektowania programów sterujących metodą manualną, przy wykorzystaniu takich narzędzi jak programowanie dialogowe czy graficzne

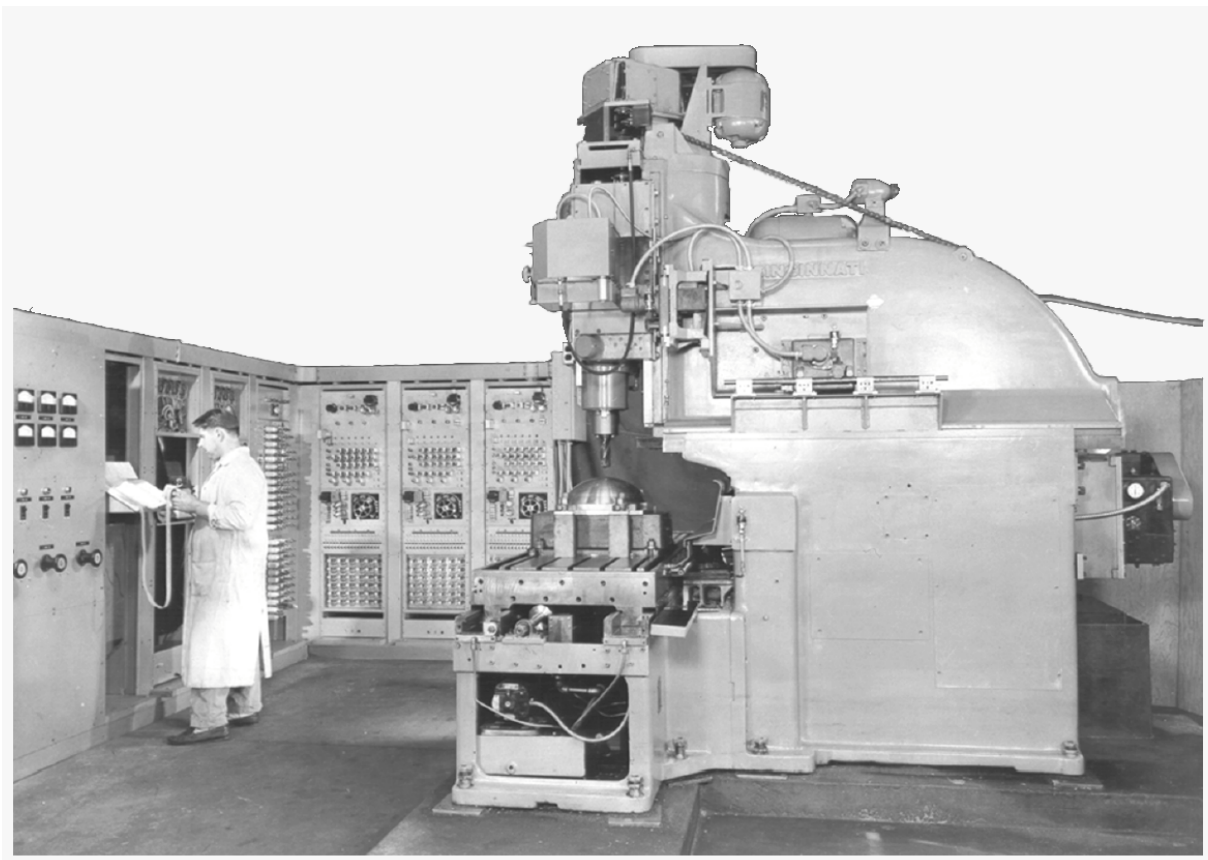
programowanie ciągów konturowych. Szerzej możliwości programu opisano w dołączonej do niego dokumentacji. Wersja ProgMastera, przeznaczona do celów edukacyjnych, dołączona została w formie załącznika.

Autor

1. ISTOTA FUNKCJONOWANIA STEROWANIA NUMERYCZNEGO

1.1. Wprowadzenie

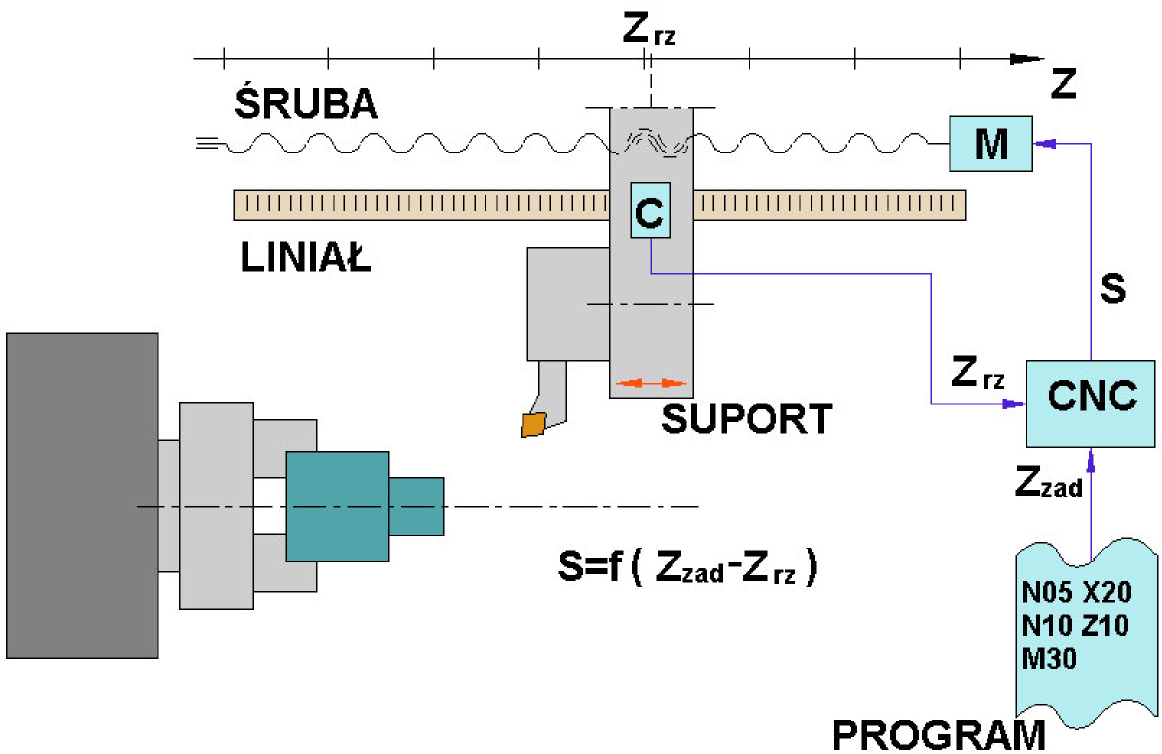
Historia obrabiarek ze sterowaniem numerycznym (NC, ang. *Numerical Control*) przekroczyła już 50 lat – pierwsza powstała w MIT w Bostonie w roku 1953 (Rys. 1). Przez pół wieku znacznie zmieniły się cechy zewnętrzne obrabiarek NC, ich kinematyka i możliwości obróbkowe, jednak idea funkcjonowania pozostała bez zmian. Jedyną istotną zmianą było zastosowanie na początku lat 70-tych komputera jako jednostki wykonawczej (CNC, ang. *Computer Numerical Control*), co znacznie zwiększyło funkcjonalność układów sterowania i spowodowało ich upowszechnienie, nie tylko w sterowaniu obrabiarek. Dzisiaj coraz częściej używa się pojęcia NC w stosunku do układów CNC, choć konstrukcyjnie różnią się one od układów NC starszych generacji, coraz rzadziej pracujących w przemyśle.



Rys. 1. Pierwsza obrabiarka ze sterowaniem numerycznym (NC)

Z punktu widzenia automatyki sterowanie CNC jest układem automatycznej regulacji programowej (stąd używane w języku polskim pojęcie „sterowanie” nie jest w pełni poprawne), pracującym w zamkniętej pętli sprzężenia zwrotnego (Rys. 2). Wartość zadana położenia (Z_{zad}) elementów ruchomych obrabiarki (np. suportu) w danej osi sterowanej numerycznie (Z) jest wyznaczana na podstawie programu. Następnie jest ona porównywana z wartością rzeczywistą położenia (Z_{rz}), mierzoną przez przetwornik pomiarowy (C). Na podstawie różnicy pomiędzy wartością zadaną a rzeczywistą położenia w osi SN układ sterowania (CNC) generuje sygnał sterujący

(S), kierowany do napędu osi (M), korygując tym samym jej położenie aż do uzyskania zerowej różnicy pomiędzy wartością zadaną a rzeczywistą położenia osi ($Z_{rz} \rightarrow Z_{zad}$).



Rys. 2. Schemat ideowy sterowania numerycznego NC

Istotą funkcjonowania obrabiarek CNC jest zatem przyjęcie założenia o istnieniu pewnego układu współrzędnych, w którym odbywa się sterowanie. Jest to najprostszy sposób na określanie **względnych** położeń narzędzia i przedmiotu obrabianego, wymaganych dla przeprowadzenia obróbki i uzyskania odpowiednich jej rezultatów. Pojęcie **numeryczny** należy więc obecnie kojarzyć ze współrzędnymi (o wartościach liczbowych, numerycznych). Należy jednak pamiętać, że źródłem nazwy „numeryczny" była postać programu sterującego, opisana w postaci kodów numerycznych (np. ASCII, ISO, EIA).

Na podstawie powyższych informacji można podać dwie najważniejsze cechy układów sterowania CNC:

➔ są to układy **sterowania programowego** – program opisuje zarówno parametry technologiczne obróbki (posuwy, prędkości skrawania, chłodzenie, itp.) jak i geometryczne (położenia zespołów ruchomych obrabiarki w trakcie obróbki);

➔ są to układy o **elastycznej postaci programu sterującego –** wymóg sterowania programowego jest warunkiem koniecznym, ale nie wystarczającym. Warunkiem tym jest taka postać programu sterującego, aby łatwo i szybko można było ją zmodyfikować (np. w celu usunięcia błędów lub zmiany wymiarów obrabianego przedmiotu). Warunek ten nie jest możliwy do spełnienia w takich układach sterowania programowego, jak np. sterowanie krzywkowe. Elastyczna postać programu sterującego predestynuje zatem obrabiarki CNC do produkcji o charakterze średnio i małoseryjnym (choć nie wyklucza wielkoseryjnej

i masowej), dominującej w dzisiejszym przemyśle. Elastyczność obrabiarek CNC to główna przyczyna ich szerokiego stosowania.

Przez **program sterujący** w układach CNC rozumie się zatem plan zamierzonej pracy obrabiarki, mającej na celu wykonanie przedmiotu o żądanych kształtach, wymiarach i chropowatości powierzchni. Składa się z następujących informacji, zapisanych w postaci alfanumerycznej:

➔ Geometrycznych, dotyczących kształtów i wymiarów, obejmujących opis toru ruchu narzędzi;

➔ Technologicznych, dotyczących warunków obróbki: narzędzia, prędkość skrawania i posuw, pomocnicze.

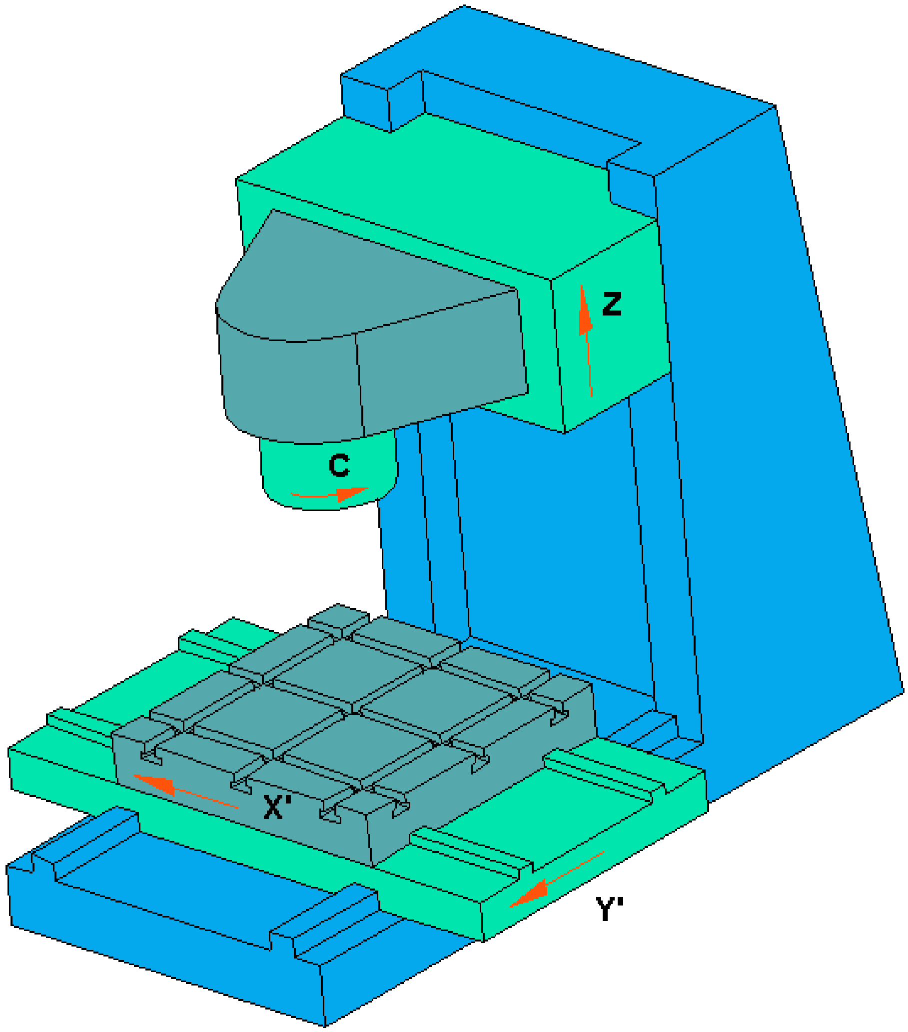
Informacje technologiczne na ogół są konsekwencją planu procesu, ustalającego wykaz zabiegów, narzędzia w nich uczestniczące, warunki ich pracy itp. W dużym stopniu wynikają one także z doświadczenia programisty. Znacznie trudniejsze jest sprecyzowanie części geometrycznej programu sterującego. Jest to w znacznym stopniu uwarunkowane rodzajem obróbki jak i informacjami zawartymi w dokumentacji konstrukcyjnej przedmiotu obrabianego. Duży wpływ mają także możliwości samego układu sterowania – dostępne sposoby wyrażania współrzędnych, dostępne cykle obróbkowe, kompensacja promienia narzędzia itp. W przypadku prostej obróbki (np. toczenie) zapis programu sterującego może w całości odbyć się metodą ręczną lub w tylko niewielkim stopniu wspomaganą komputerowo, często ograniczając się do symulacji programu. Dla obróbki powierzchni swobodnych stosuje tylko automatyczne generowanie programu sterującego przy pomocy **systemów CAM** (bardzo obszerne programy wymagające dużego nakładu obliczeniowego). Niezależnie od metody programowania znajomość struktury programu wydaje się być niezbędny (np. do zdefiniowania postprocesorów w systemach CAM). Dokładny opis struktury programu i jego elementów składowych omówiono zatem w dalszej części niniejszego skryptu.

Programowanie polega więc przede wszystkim na zapisie ruchów wykonywanych przez obrabiarkę w trakcie obróbki. Ruchy te mogą mieć dwojaki charakter:

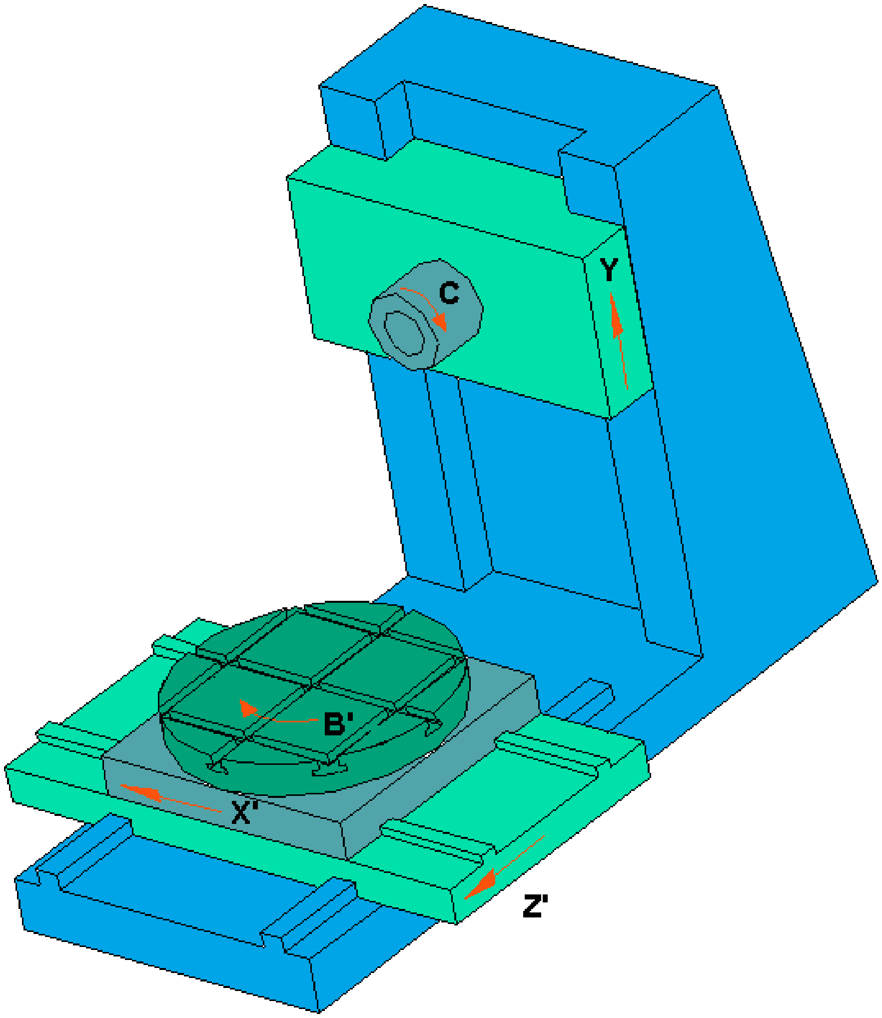
➔ sterowane w sposób ciągły (ciągły pomiar położenia, ciągłe sterowanie napędem), są one ogólnie nazywane **osiami sterowanymi numerycznie** (SN). Są to ruchy zarówno liniowe (oznaczane symbolami X, Y, Z,....) jak i obrotowe (oznaczane symbolami A, B, C,...). Stanowią one zasadniczą część programu sterującego a funkcje je obsługujące stanowią standard języka układu sterowania, zaprojektowany przez producenta układu sterowania.

➔ sterowane w sposób dyskretny (typu włącz – wyłącz, obroty w lewo – obroty w prawo itp.). Ich realizacja ma w programie sterującym charakter pomocniczy (np. obsługa silnika pompki chłodziwa, zamykanie – otwieranie podtrzymki, uruchamianie podajnika pręta, wymiana palet itp.) dlatego są obsługiwane przez specjalną grupę funkcji, zwanych **pomocniczymi**. Część funkcji pomocniczych stanowi standard języka układu sterowania (opis w dokumentacji języka), większość jednak jest implementowane przez producenta obrabiarki w zależności od fizycznych urządzeń na niej zainstalowanych (opis w dokumentacji techniczno-ruchowej obrabiarki).

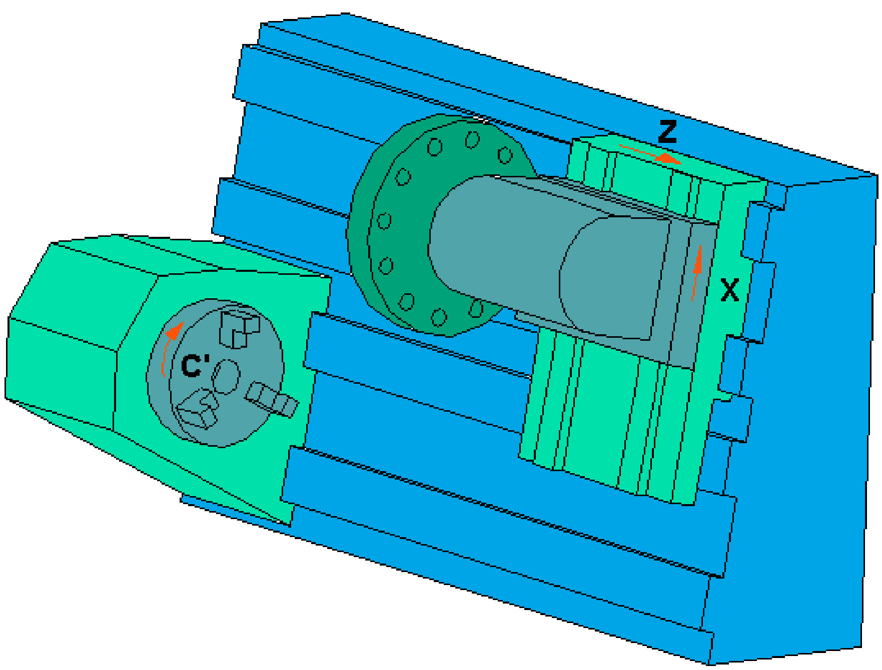
Z **osią sterowaną numerycznie** związany jest zawsze **oddzielny napęd** (silnik, siłownik) jak i **układ pomiarowy**. Te cechy odróżniają obrabiarki CNC od innych rodzajów obrabiarek, gdzie napęd najczęściej jest scentralizowany. Na Rys. 3, Rys. 4 i Rys. 5 pokazano typowe obrabiarki CNC wraz z układem i typowymi oznaczeniami osi sterowanych numerycznie.



Rys. 3. Układ i oznaczenia osi sterowanych numerycznie dla frezarki pionowej



Rys. 4. Układ i oznaczenia osi sterowanych numerycznie dla frezarki poziomej



Rys. 5. Układ i oznaczenia osi sterowanych numerycznie dla tokarki

Inne cechy, charakterystyczne dla obrabiarek CNC, to (są one szerzej omawiane w innych publikacjach):

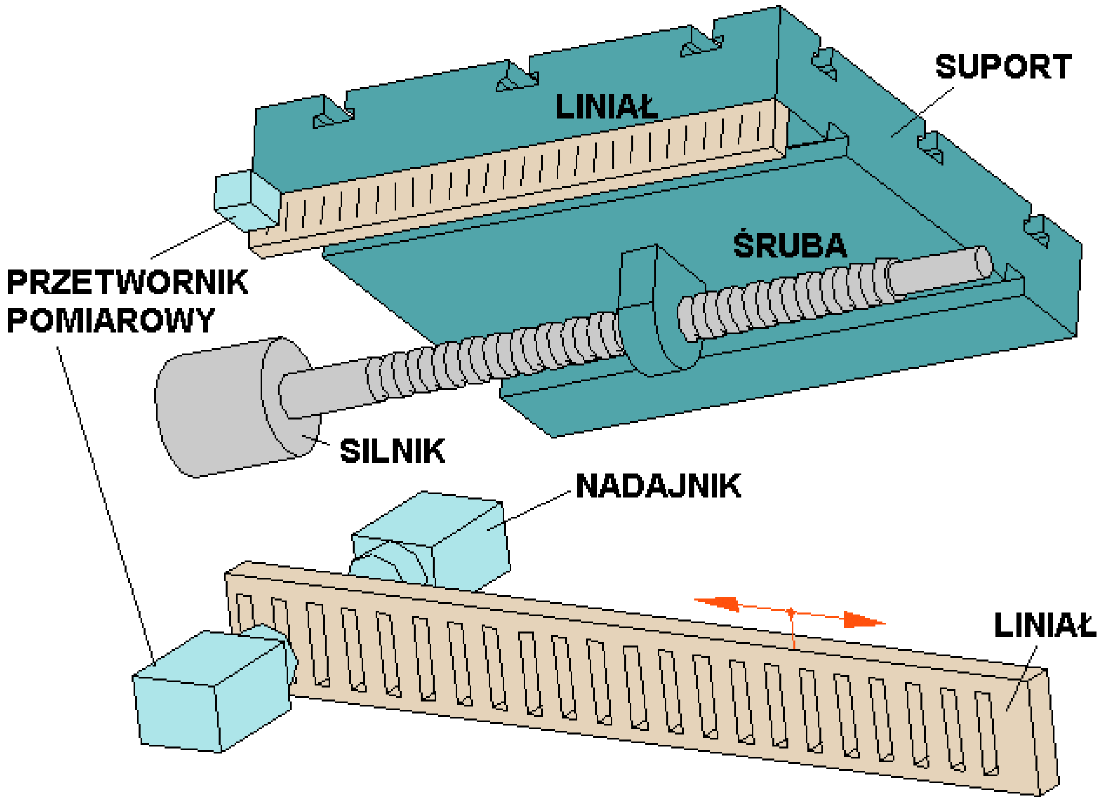
- ➔ Bezstopniowa regulacja prędkości obrotowej i posuwów;
- ➔ Napęd przenoszony za pomocą śrub tocznych;
- ➔ Eliminowanie prowadnic ślizgowych na rzecz tocznych;
- ➔ Eliminowanie przekładni zębatych;
- ➔ Kompaktowa konstrukcja o zamkniętej przestrzeni roboczej;
- ➔ Konstrukcja modułowa o elastycznie dobieranej konfiguracji elementów składowych;
- ➔ Mała podatność statyczna i dynamiczna;
- ➔ Automatyczny nadzór i diagnostyka;
- ➔ Duża moc (jako suma mocy poszczególnych napędów);
- ➔ Osiąganie znacznych wartości parametrów obróbki (np. duże prędkości obrotowe);
- ➔ Obróbka równoległa z wykorzystaniem wielu wrzecion i/lub suportów narzędziowych;
- ➔ Złożona kinematyka pracy (uchylne głowice narzędziowe, stoły obrotowo-uchylne, obróbka pięcioosiowa, obrabiarki o strukturze równoległej);
- ➔ Magazyny narzędziowe z automatyczną wymianą narzędzi;
- ➔ Systemy narzędziowe z narzędziami składanymi;
- ➔ Nowoczesne materiały narzędziowe;
- ➔ Automatyczny pomiar narzędzi;
- ➔ Kodowanie narzędzi;
- ➔ Automatyczna wymiana przedmiotu obrabianego;
- ➔ Automatyczny pomiar przedmiotu obrabianego;
- ➔ Automatyczne usuwanie wiórów.

Występowanie powyższych cech w konkretnej obrabiarce często zależy od tego, w jak dużym stopniu jest ona przystosowana do pracy autonomicznej (bez obsługi człowieka).

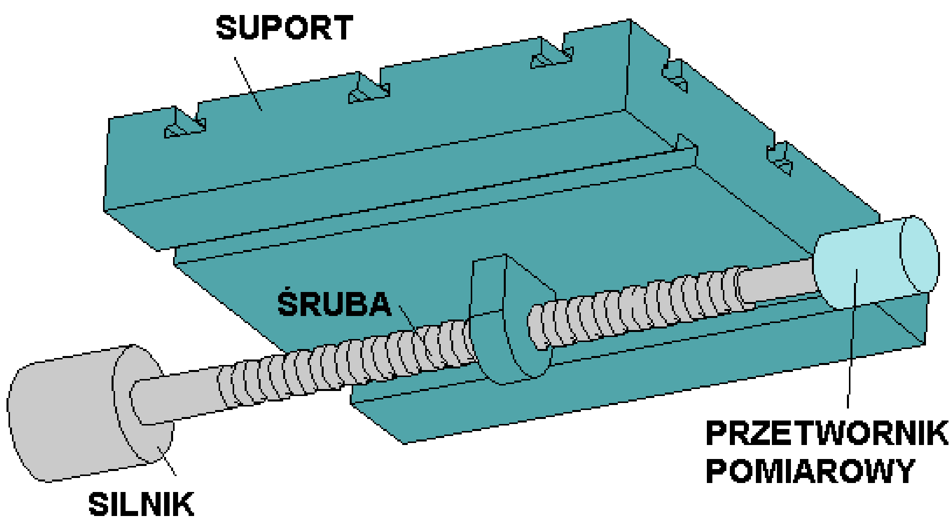
1.2. Pomiary położenia w osiach sterowanych numerycznie

Jak wspomniano w poprzednim rozdziale, pomiar położenia to warunek konieczny poprawnego funkcjonowania osi sterowanej numerycznie. Układy pomiaru położenia można podzielić na dwie grupy:

- ➔ z **bezpośrednim pomiarem położenia** – czujnik pomiarowy mierzy wprost położenie danego elementu obrabiarki (np. suportu) – na Rys. 6 pokazano pomiar położenia suportu liniowego za pomocą liniału i przetwornika optoelektronicznego.
- ➔ z **pośrednim pomiarem położenia** – czujnik pomiarowy mierzy pewną wielkość pośrednio związaną z położeniem danego elementu obrabiarki, na podstawie której to położenie jest wewnętrznie obliczane przez układ pomiarowy. Dla przykładu na Rys. 7 pokazano pomiar pośredni położenia suportu liniowego za pomocą kątowego przetwornika pomiarowego, gdzie na podstawie położenia kątowego śruby przemieszczającej suport i znajomości jej skoku jest obliczane jego położenie.



Rys. 6. Pomiar bezpośredni położenia



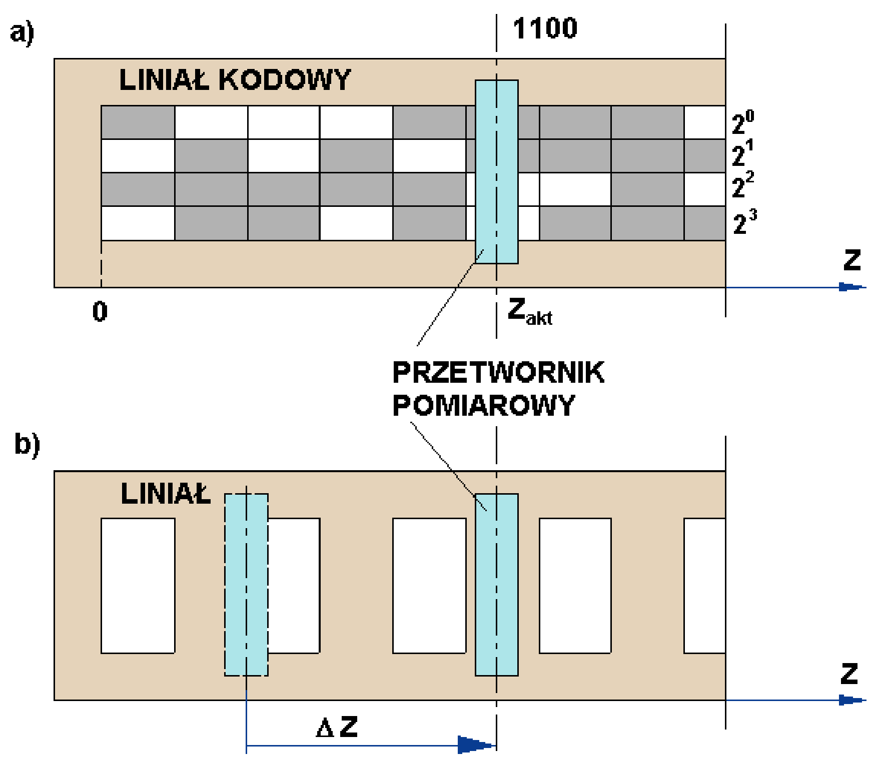
Rys. 7. Pomiar pośredni położenia

Ze względu na charakter pracy układy pomiaru położenia można podzielić na dwa rodzaje:

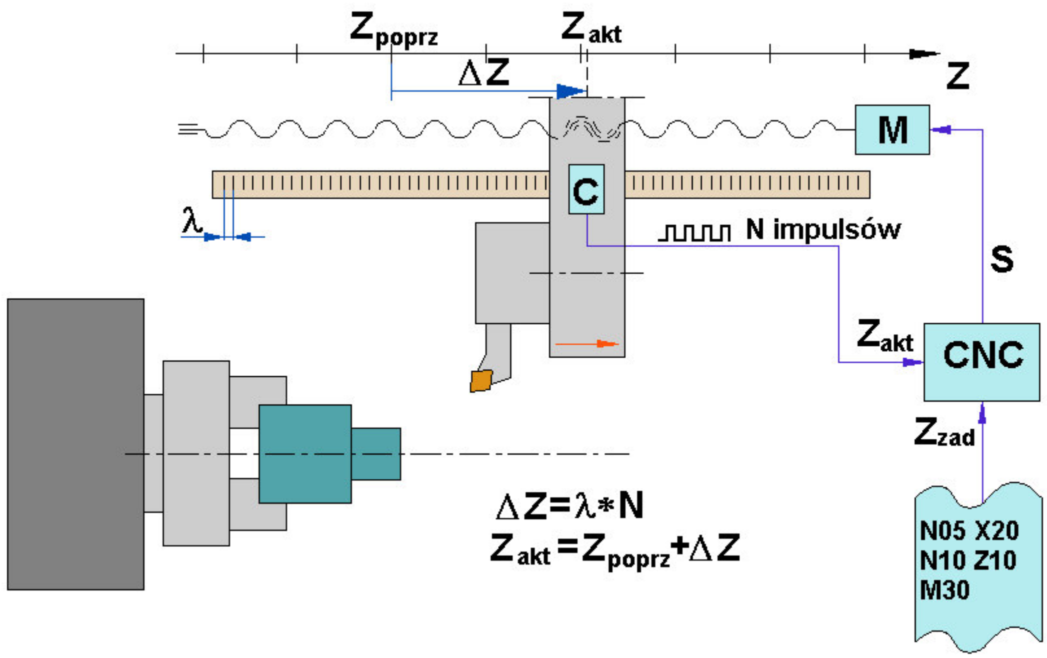
➔ **absolutne układy pomiaru położenia** (Rys. 8a) – sygnał wyjściowy przetwornika pomiarowego wprost zawiera informację (najczęściej w postaci zakodowanej, np. w kodzie binarnym, Graya, Wattsa) o mierzonym położeniu. Działanie takie pozwala na stałe ustalenie punktu zerowego osi sterowanej numerycznie, działanie układu pomiarowego nie wymaga dodatkowych czynności po włączeniu zasilania

obrabiarki. Wadą układów absolutnego pomiaru położenia jest bardziej skomplikowana budowa niż układów przyrostowych (a więc i wyższy koszt) dlatego są one rzadko stosowane.

➔ **przyrostowe (inkrementalne) układy pomiaru położenia** (Rys. 8b) – sygnał wyjściowy przetwornika jest ciągiem impulsów (umownie można je nazwać jako 0 i 1), które wprost nie niosą informacji o absolutnym położeniu, ale o **przyrostowej zmianie położenia** w osi sterowanej numerycznie. Zasadniczo układ pomiarowy działa jako licznik impulsów (Rys. 9). Na podstawie stanu licznika (liczby impulsów *N*) oraz znajomości wartości działki elementarnej λ liniału pomiarowego obliczana jest zmiana położenia (ΔX), a położenie rzeczywiste w osi SN (X_{rz}) jest sumą wartości współrzędnej położenia poprzedniego (X_0) i zmiany położenia w osi SN (ΔX). Wadą tych układów jest „płynne" położenie punktu zerowego, co powoduje konieczność wykonywania tzw. zerowania osi po włączeniu zasilania obrabiarki. Tym niemniej ze względu na prostszą konstrukcję i niższy koszt są one powszechnie stosowane w obrabiarkach CNC. Zasada pracy układów przyrostowych opiera się na wykorzystaniu liniału z naprzemiennie położonymi polami o zmiennej charakterystyce optycznej, indukcyjnej czy pojemnościowej. Przesuwający się względem liniału przetwornik pomiarowy przetwarza zmiany strumienia światła, indukcyjności czy pojemności elektrycznej na zmienny sygnał wyjściowy, najczęściej w postaci napięcia elektrycznego, przekazywanego do liczników przetwarzających ten sygnał (Rys. 6).



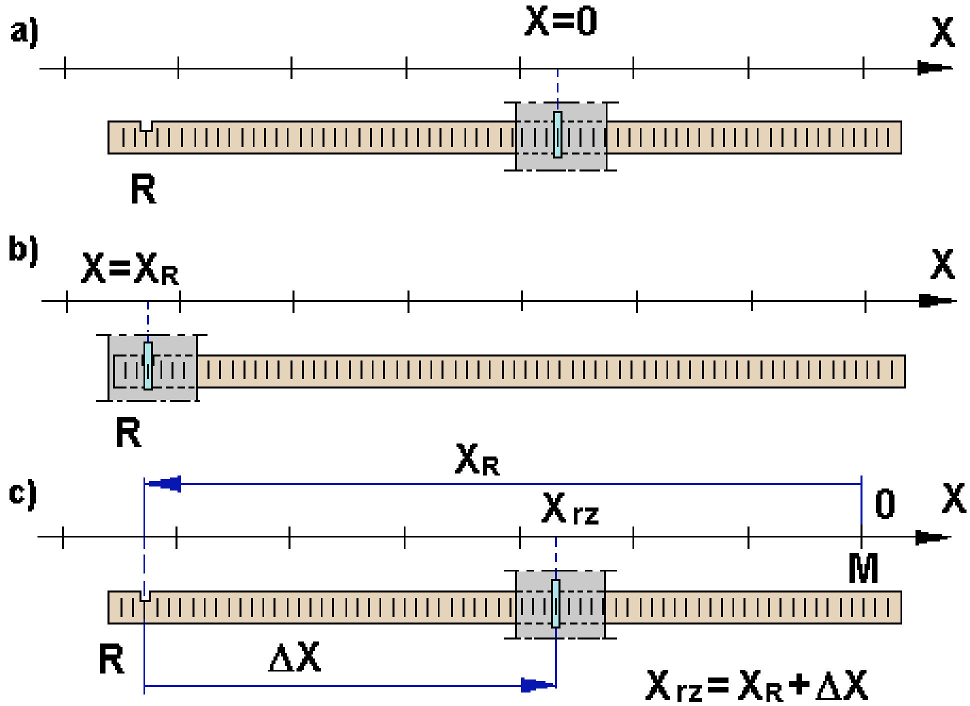
Rys. 8. Absolutny (a) i przyrostowy (b) pomiar położenia



Rys. 9. Przyrostowy pomiar położenia w układzie sterowania CNC

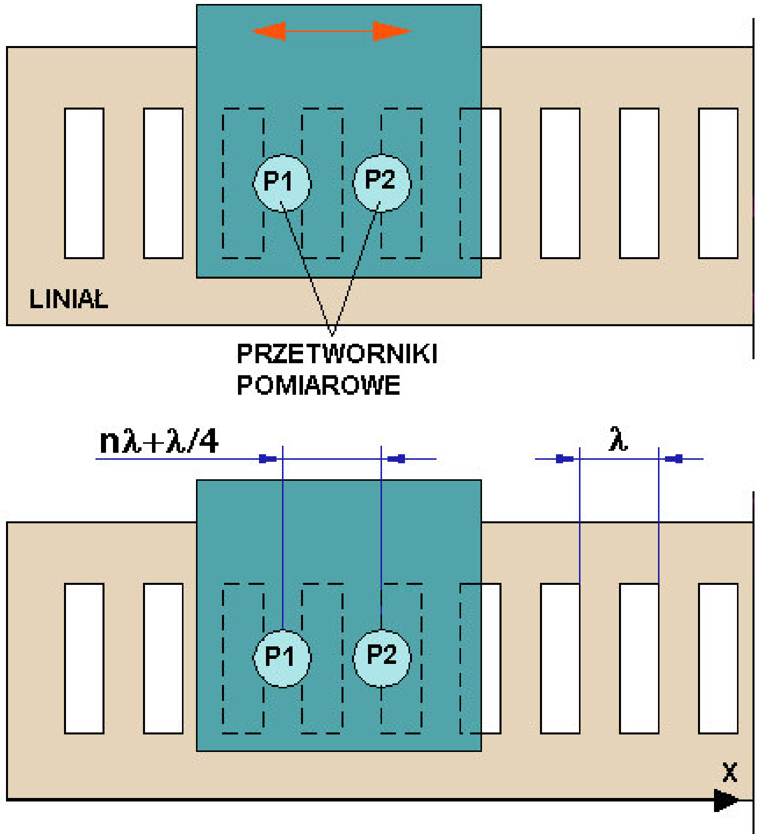
Stosowanie przyrostowych układów pomiarowych wiąże się z dwoma istotnymi problemami:

- ➔ nie posiadają stałego położenia punktu zerowego, co wyklucza powtarzalność pracy obrabiarki CNC – po każdym włączeniu zasilania punkt zerowy osi SN znajdowałby się w innym miejscu (zależnym od aktualnego położenia zespołów ruchomych obrabiarki) z uwagi na automatyczne zerowanie liczników impulsów pomiarowych; aby wyeliminować tą wadę na liniale pomiarowym nanosi się specjalny znacznik (może to być np. wyłącznik drogowy) o stałym położeniu (a więc i stałej wartości absolutnej współrzędnej). Nosi on nazwę **punktu referencyjnego** (oznaczany jako **R**). Położenie tego punktu jest mierzone przez producentów obrabiarek od umownie przyjętego punktu zerowego danej osi SN (zwanego punktem maszynowym **M**) i wprowadzane do pamięci układu sterowania. Po każdorazowym uruchomieniu obrabiarki, kiedy położenia jej zespołów ruchomych są przypadkowe (Rys. 10a) pierwszą czynnością jest przemieszczenie ich do punktu referencyjnego (Rys. 10b), co nazywane jest **najazdem na punkt referencyjny**, **zerowaniem**, **bazowaniem**. W tym położeniu następuje zerowanie liczników odczytujących położenie i wprowadzanie do nich wartości odpowiadających pobranym z pamięci położeniom referencyjnym (Rys. 10c). Tym samym jest określone stałe, niezmienne w czasie położenie punktu zerowego osi SN, umożliwiające stosowanie absolutnego układu współrzędnych.

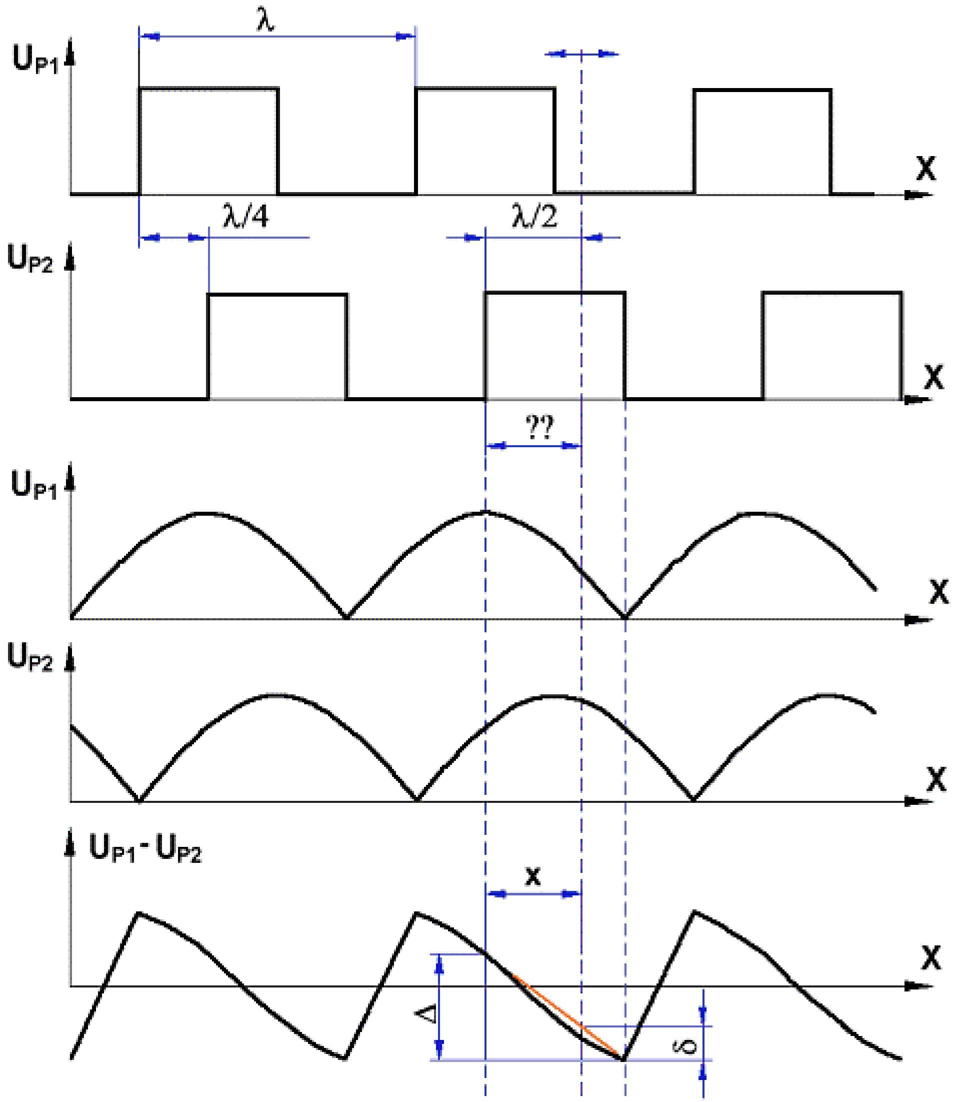


Rys. 10. Zasada najazdu na punkt referencyjny osi SN: a - po włączeniu obrabiarki, b - najazd na punkt referencyjny **R**, c - ustawienie punktu zerowego w punkcie maszynowym **M**

➔ przyrostowe układy pomiarowe posiadają stosunkowo dużą wartość podziałki elementarnej λ (rzędu 0,1÷0,01 mm), co powodowałoby małą dokładność odczytu wartości położenia; dodatkowo oprócz odczytu zmiany wartości położenia w osi SN powinna być również podana informacja o kierunku tej zmiany (znak wartości *ΔX* na Rys. 10c). W tym celu stosuje się zwielokrotnione układy przetworników pomiarowych (minimum dwa) – Rys. 11. Dodatkowo ważne jest ich wzajemne położenie, przesunięte w fazie o ¼ długości działki elementarnej λ. Efekt tego przesunięcia jest widoczny na wykresach sygnałów wyjściowych przetworników, zamieszczonych na Rys. 12. Kierunek przesunięcia elementu ruchomego jest wykrywany poprzez analizę kolejności impulsów z przetworników (Rys. 12a). Jeżeli w sytuacji jak na rysunku przemieszczenie nastąpi w lewo to pierwszy zawsze wystąpi impuls z przetwornika P2, w przeciwnym przypadku – z przetwornika P1. Z kolei zwiększona dokładność odczytu położenia związana jest z wykorzystaniem różnicy sygnałów z obu przetworników (Rys. 12b). Zbliżona do liniowej charakterystyka tego sygnału za pomocą interpolacji pozwala mierzyć położenie z dokładnością 10, a nawet 100 razy większą, niż podziałka elementarna λ układu pomiarowego.



Rys. 11. Zwielokrotniony układ przetworników pomiarowych przy przyrostowym pomiarze położenia w osi SN



Rys. 12. Przebiegi sygnałów na przetwornikach pomiarowych w układzie zdwojonym:
a - teoretyczne, b - rzeczywiste

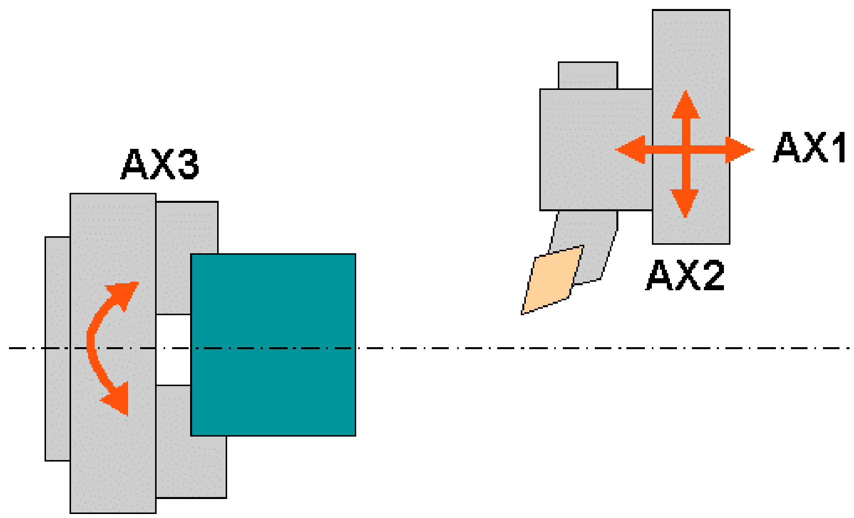
1.3. Układy współrzędnych

Podstawą do programowania jest zdefiniowanie układu współrzędnych, dzięki któremu możliwe jest zadawanie współrzędnych położeń elementów ruchomych obrabiarki CNC. W rzeczywistości na każdej obrabiarce istnieje wiele różnych układów współrzędnych. Na szczególną uwagę zasługują trzy z nich:

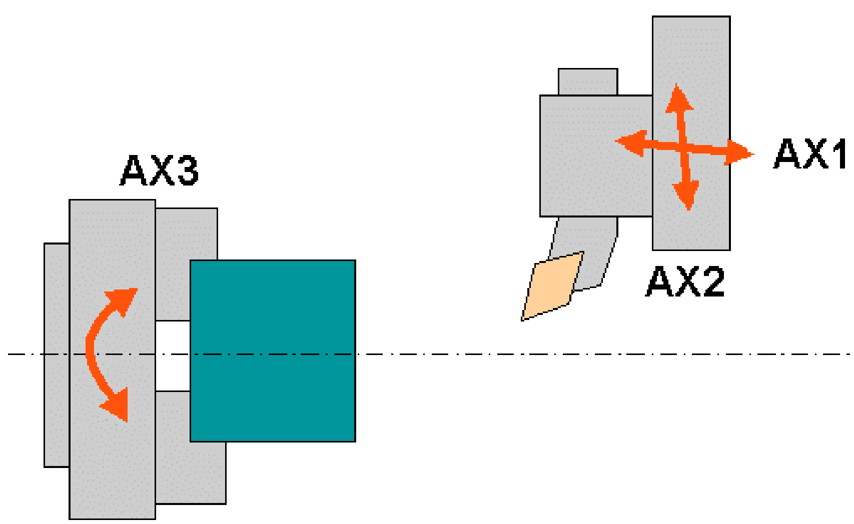
- ➔ Maszynowy układ współrzędnych;
- ➔ Bazowy (podstawowy) układ współrzędnych;
- ➔ Układ współrzędnych przedmiotu.

1. **Maszynowy układ współrzędnych** (**MKS,** niem. *Maschinen Koordinaten Systeme*) – układ współrzędnych zbudowanych z osi sterowanych numerycznie obrabiarki lub innego urządzenia sterowanego numerycznie (Rys. 3, Rys. 4, Rys. 5), zdefiniowany przez prowadnice, łożyskowanie i inne elementy konstrukcji obrabiarki. W tym układzie odbywa się sterowanie, tylko w tym układzie osie posiadają niezależne napędy i układy pomiarowe. Współrzędne zadane w innych

układach współrzędnych są przeliczane na układ maszynowy przez sterownik CNC i na odwrót – bieżące współrzędne maszynowe są przeliczane na inne układy współrzędnych. Układ maszynowy może być układem prostokątnym, walcowym, sferycznym lub o złożonym charakterze (np. w robotach). Układ maszynowy jest odniesiony do konstrukcji konkretnej obrabiarki i nie jest objęty normami. W układzie maszynowym są podane współrzędne punktu referencyjnego (**R**), punkty wymiany narzędzi, punkty wymiany palet itp. Osie maszynowego układu współrzędnych mogą być oznaczane kolejnymi cyframi (1, 2, 3.., AX1, AX2,....) lub oznaczeniami podobnymi do osi pozostałych układów współrzędnych (np. X, Y, Z, X1, Y1, Z1) – Rys. 13. Układ maszynowy jest układem rzeczywistym, tj. obarczonym różnego rodzaju błędami wykonawczymi – nieprostoliniowość osi, nieprostopadłość osi, błędy podziałki itp. – Rys. 14. Obróbka przy wykorzystaniu takiego układu wyklucza uzyskanie wysokiej jakości produktów. Jeżeli jednak wspomniane błędy są znane przez układ sterowania to drogą programową mogą zostać skompensowane (nie jest to możliwe na obrabiarkach konwencjonalnych).

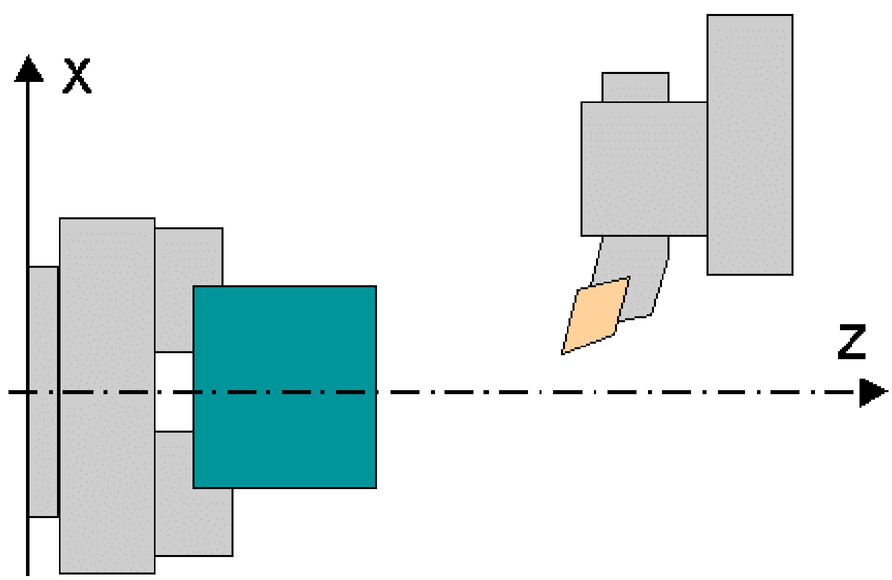


Rys. 13. Maszynowy układ współrzędnych na przykładzie tokarki



Rys. 14. Rzeczywisty układ osi maszynowych (przykład)

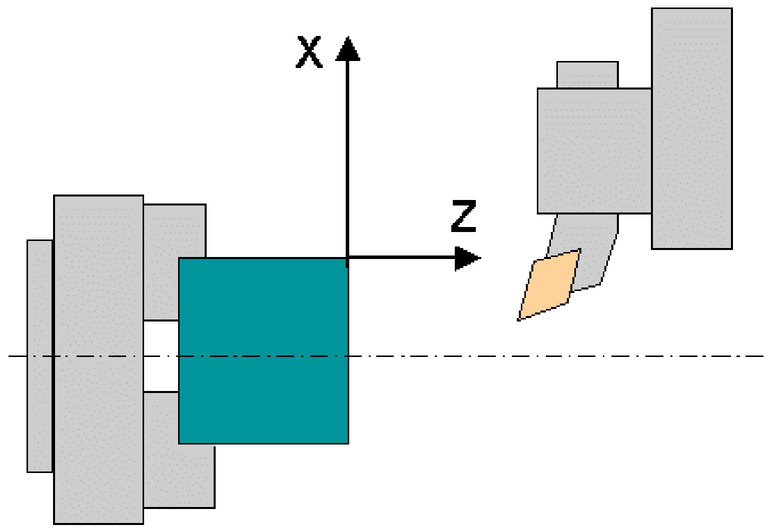
2. **Bazowy (podstawowy) układ współrzędnych** (**BKS**, niem. *Basis Koordinaten Systeme*) – **prostokątny, prawoskrętny układ współrzędnych**, stanowiący podstawę do programowania (Rys. 15). Jest odniesiony do przedmiotu zamocowanego na obrabiarce, traktowanego jako nieruchomy, przy poruszającym się narzędziu (zakłada się względny ruch narzędzia względem przedmiotu obrabianego). Jest związany z układem maszynowym poprzez transformacje kinematyczne, odwzorowujące układ bazowy na osie maszynowe (np. osie sferycznego układu współrzędnych robota, transformacja we frezarce 5-osiowej itp.). W najprostszym przypadku układ bazowy (tokarki, frezarki) jest tożsamy z układem maszynowym – brak transformacji kinematycznych. W tym układzie są definiowane korektory narzędziowe, wyznaczana jest kompensacja promienia narzędzia, wyznaczane są transformacje układu przedmiotu, ustawiane są granice obszaru obróbki oraz wykonywane są wszystkie obliczenia toru ruchu narzędzia. Służy on zatem przede wszystkim układowi CNC. Wymagania stawiane układowi bazowemu są unormowane, szerzej zostanie to omówione w dalszej części skryptu. Układ bazowy jest zdefiniowany przez producenta układu sterowania, zadaniem producenta obrabiarki, którą on steruje, jest powiązanie układu maszynowego i bazowego (poprzez tzw. dane maszynowe).



Rys. 15. Bazowy układ współrzędnych

3. **Układ współrzędnych przedmiotu** (**WKS**, niem. *Werkstück Koordinaten Systeme*) – prostokątny, prawoskrętny układ współrzędnych, związany z przedmiotem obrabianym, służący do programowania obróbki, zapisanej w postaci programu sterującego (Rys. 16). Jest przekształconym układem bazowym poprzez definicję tzw. FRAMES, będących matematycznymi formułami matematycznymi, przekształcającymi układy współrzędnych z wykorzystaniem czterech podstawowych działań:
   - Translacji o wektor;
   - Obrotu wokół osi;
   - Symetrii osiowej (odbicia lustrzanego);
   - Skalowania osi.

Zapis matematyczny tych działań jest realizowany z wykorzystaniem rachunku macierzowego. Wybór układu współrzędnych przedmiotu zależy od sposobu jego wymiarowania, możliwe jest użycie w jednym programie sterującym kilku różnych układów współrzędnych przedmiotu.

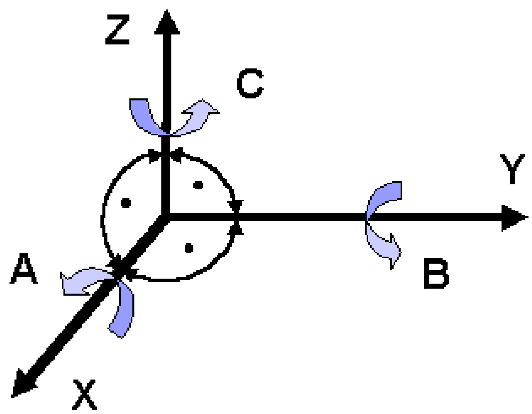


Rys. 16. Układ współrzędnych przedmiotu (WKS)

1.4. Definicja układów współrzędnych

Przy definicji układów współrzędnych (dotyczy to przede wszystkim układu bazowego i przedmiotu) są stosowane pewne zasady, pozwalające na ich unifikację. Zasady te są zawarte w normach. W Polsce jest to norma M-55251, oparta na normie DIN 66217 oraz na normach międzynarodowych (ISO). Podstawowe zasady definiowania układów współrzędnych są następujące:

- ➔ Osie układu współrzędnych są odniesione do przedmiotu obrabianego;
- ➔ Przyjmuje się przedmiot obrabiany za nieruchomy, porusza się tylko narzędzie (układ współrzędnych może się przemieszczać z przedmiotem obrabianym);
- ➔ Podstawowym układem jest prostokątny, prawoskrętny układ współrzędnych (Rys. 17);
- ➔ Za podstawowe przyjmuje się nazwy osi liniowych X, Y i Z. W szczególnych przypadkach osie mogą przyjmować inne nazwy, np. U, V, W, P, Q, R;
- ➔ Sterowane numerycznie osie obrotowe przyjmują nazwy A, B, C. Są one związane z osiami liniowymi (A obrót wokół X, B wokół Y, C wokół Z). Zwroty dodatnie przyjmuje się zgodnie z regułą śruby prawoskrętnej;
- ➔ Jeżeli osie związane są z ruchem przedmiotu obrabianego przyjmują indeks ’ (np. X’) i zwrot przeciwny do zwrotu danej osi sterowanej numerycznie (np. X).



Rys. 17. Układ i oznaczenia osi prostokątnego, prawoskrętnego układu współrzędnych

Definicja układu współrzędnych obejmuje trzy fazy (Rys. 18):

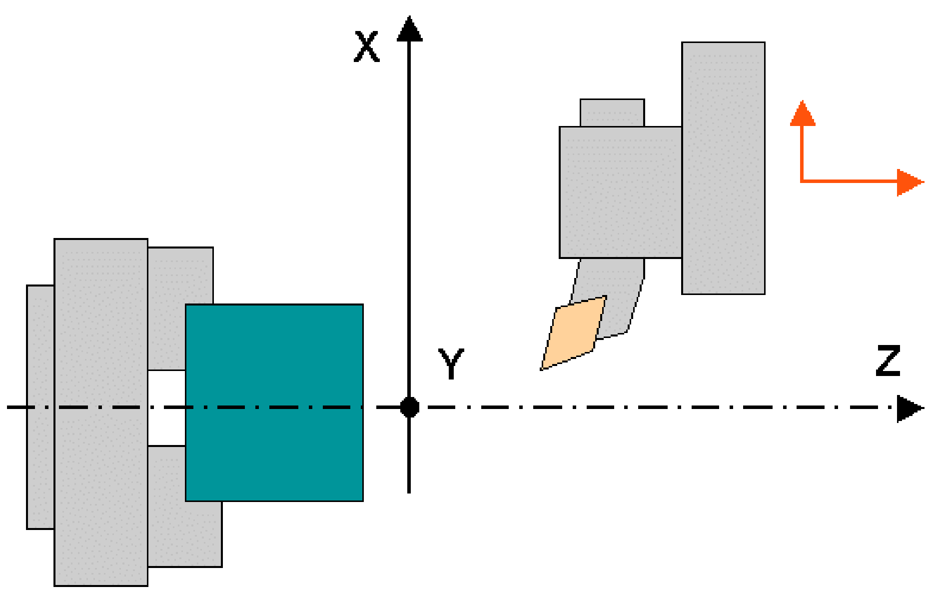
1. **Kierunki osi**:

➔ W pierwszej kolejności definiuje się kierunek osi Z, który powinien być zgodny lub pokrywać się z osią wrzeciona głównego (przedmiotowego lub narzędziowego).

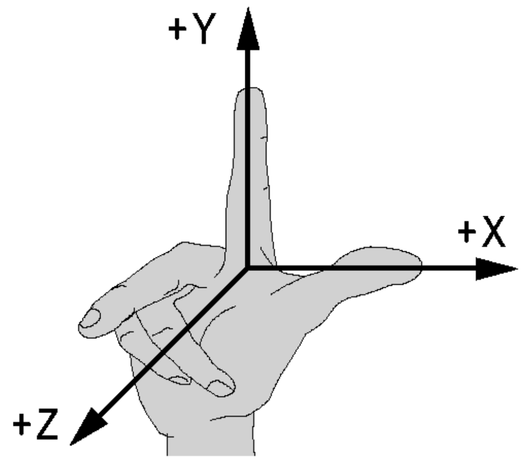
➔ W drugiej kolejności definiuje się kierunek osi X. Jest on prostopadły do kierunku osi Z i na ogół leży w płaszczyźnie równoległej do płaszczyzny mocowania przedmiotu lub prowadnic obrabiarki związanych ze stołem przedmiotowym.

➔ W trzeciej kolejności wyznacza się kierunek osi Y korzystając z właściwości prostopadłości osi układu współrzędnych.

2. **Zwroty osi**: za zwrot dodatni osi uważa się taki, z którym związany jest ruch od strony przedmiotu obrabianego (ruch do materiału wg ujemnego zwrotu osi). Zwroty osi należy tak ustalić, aby spełniały warunek prawoskrętności (reguła śruby prawoskrętnej lub prawej dłoni – Rys. 19).



Rys. 18. Definiowanie kierunków i zwrotów osi układu współrzędnych na przykładzie tokarki



Rys. 19. Zasada prawej dłoni przy wyznaczaniu zwrotów osi

3. **Punkty zerowe układu współrzędnych** i punkty, których współrzędne są kodowane w danym układzie współrzędnych: przyjmuje się pewne punkty charakterystyczne dla każdej maszyny NC, omówione w następnym rozdziale.

1.5. Punkty charakterystyczne obrabiarki

Każda obrabiarka posiada charakterystyczne punkty, odnoszące się do zdefiniowanych układów współrzędnych. Najważniejsze z nich to [PN-ISO 3002] – Rys. 20, Rys. 21:

M – punkt maszynowy (niem. *Maschinen-Nullpunkt*); punkt początku maszynowego układu współrzędnych MKS (równocześnie również układu bazowego BKS). Jego położenie jest ustalane przez producenta obrabiarki, na ogół ściśle związane z jej konstrukcją. Do niego odnoszone są pozostałe punkty charakterystyczne.

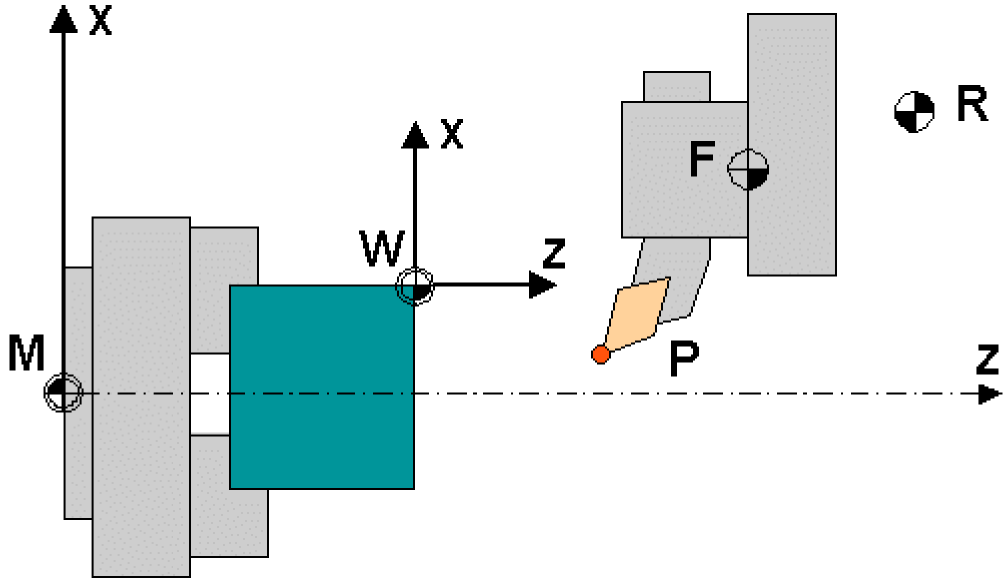
W – punkt zerowy przedmiotu (niem. *Werkstück-Nullpunkt*). Punkt początku układu współrzędnych przedmiotu WKS, ustalany w sposób dowolny przez programistę. Warunkiem poprawnej pracy obrabiarki jest wprowadzenie do układu sterowania informacji o położeniu tego punktu.

R – punkt referencyjny (niem. *Referenzpunkt*). Punkt o znanej odległości od punktu maszynowego **M**, służący do ustalenia położenia punktu początku osi układu maszynowego MKS (bazowego BKS). Jego położenie ustala producent obrabiarki.

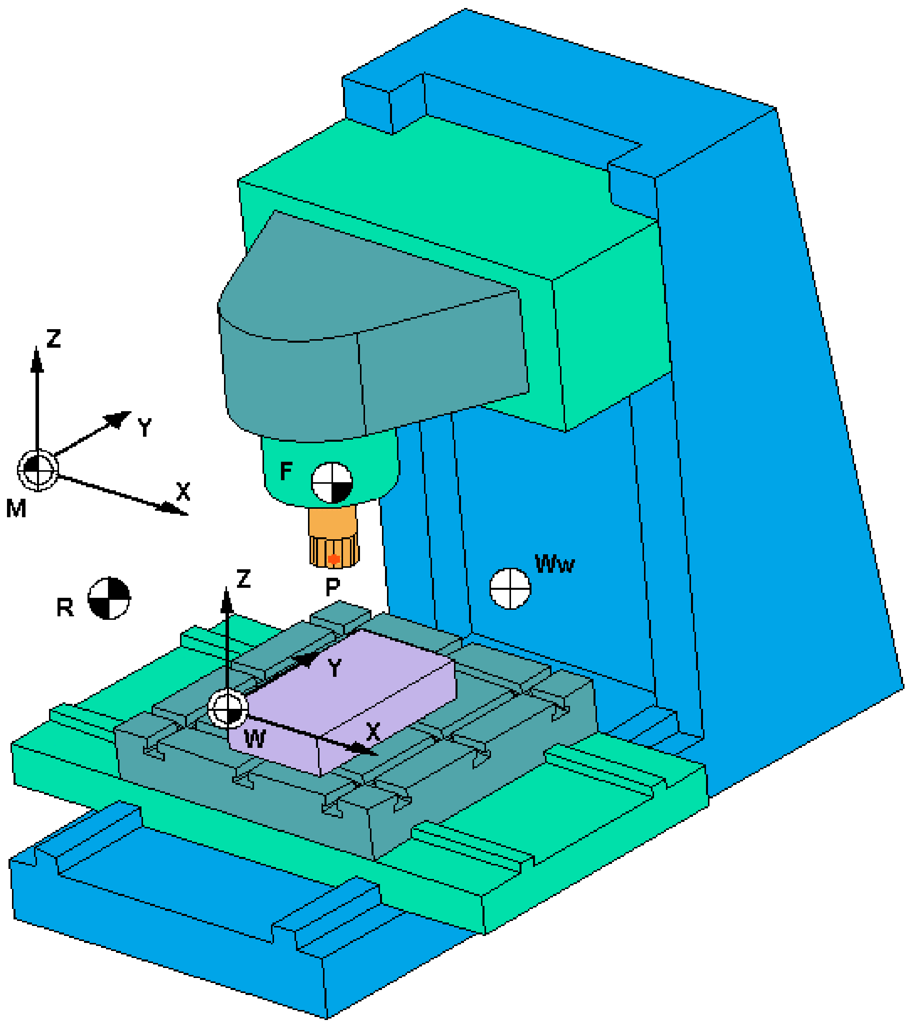
F – punkt odniesienia zespołu narzędziowego. Punkt kodowy, którego współrzędne są podawane w układzie współrzędnych MKS (BKS). Położenie tego punktu związane jest z konstrukcją obrabiarki. Dla celów programowania obróbki znajomość położenia tego punktu nie ma istotnego znaczenia co zostanie wyjaśnione w dalszych rozdziałach.

P – punkt kodowy narzędzia. Punkt, którego współrzędne są zadawane w programie sterującym. położenie tego punktu przyjmuje programista obrabiarki w zależności od rodzaju narzędzia i jego przeznaczenia. Warunkiem poprawnej pracy obrabiarki jest wprowadzenie do układu sterowania informacji o położeniu tego punktu.

Ww – punkt wymiany narzędzia (niem. *WerkzeugWechselpunkt*). W tym punkcie musi znaleźć się punkt kodowy **F** aby w sposób prawidłowy i bezpieczny dokonać wymiany narzędzia (nie jest to wymagane dla wszystkich obrabiarek).

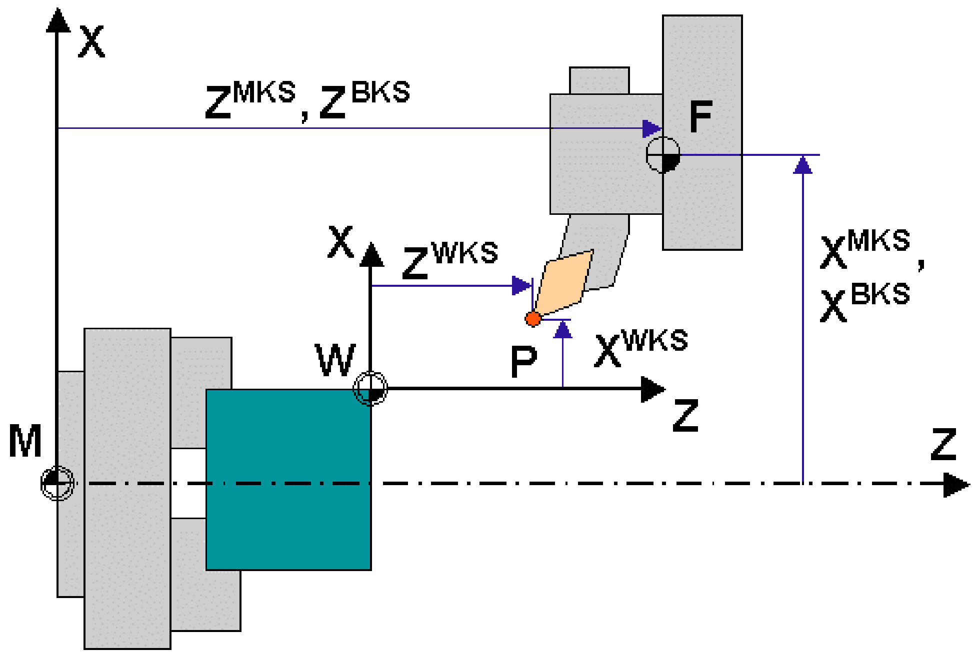


Rys. 20. Punkty charakterystyczne tokarki CNC

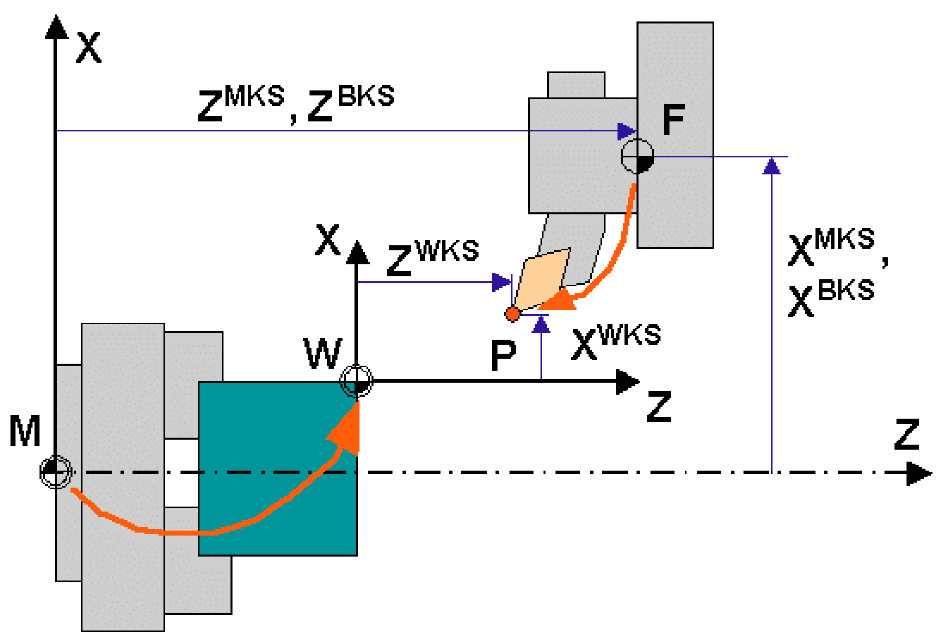


Rys. 21. Punkty charakterystyczne frezarki pionowej CNC

Korzystając z tak zdefiniowanych punktów charakterystycznych możliwe jest podanie ostatecznych definicji podstawowych układów współrzędnych obrabiarki: maszynowego, bazowego i przedmiotu (Rys. 22). Jak łatwo zauważyć, układy maszynowy i bazowy oraz przedmiotu różnią się między sobą zarówno położeniem punktu zerowego (**M** i **W**), jak i punktu kodowego (**F** i **P**), którego współrzędne są zadawane i odczytywane (Rys. 23). Dodatkowo możliwe są inne zmiany układów współrzędnych, takie jak obroty, skalowanie itp. Szerzej o przekształceniach układów współrzędnych traktują następne rozdziały.

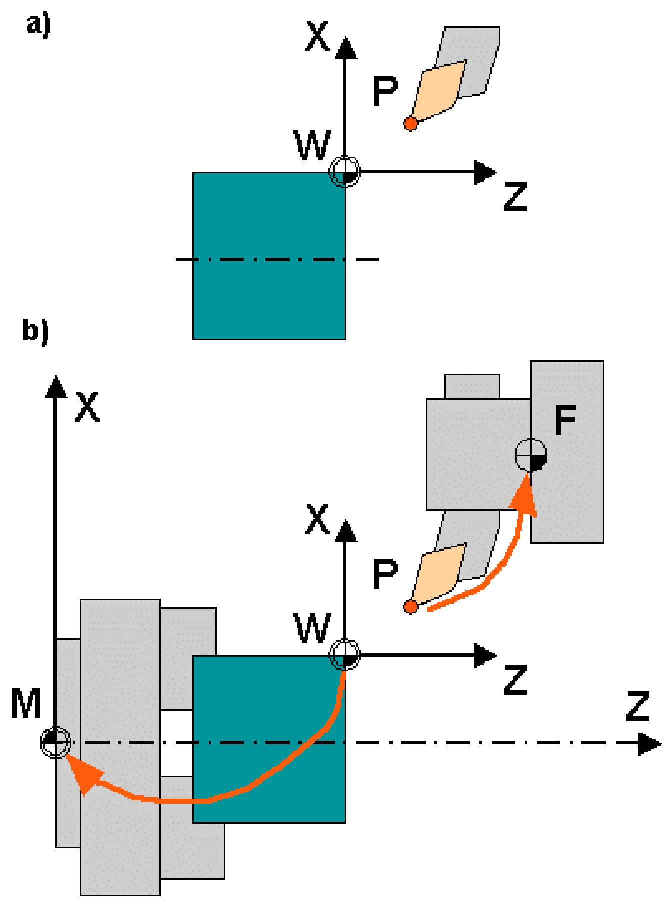


Rys. 22. Współrzędne w układzie MKS, BKS i WKS



Rys. 23. Przekształcenie układu MKS (BKS) w układ WKS

Analiza Rys. 23 pozwala na sformułowanie podstawowej zasady tworzenia programowania: program sterujący pisany w układzie przedmiotu WKS (Rys. 24a), wyizolowanym od rzeczywistych warunków jego realizacji może być wykonany na dowolnej obrabiarce (w jej unikalnym układzie maszynowym MKS) pod warunkiem poprawnego zadania wartości rejestrów przesunięć punktów zerowych i rejestrów narzędziowych – Rys. 24b. Tym samym program sterujący **jest niezależny od obrabiarki** na której będzie wykonywany – ten sam program może więc być uruchomiony na kilku różnych obrabiarkach (o różnych układach maszynowych).

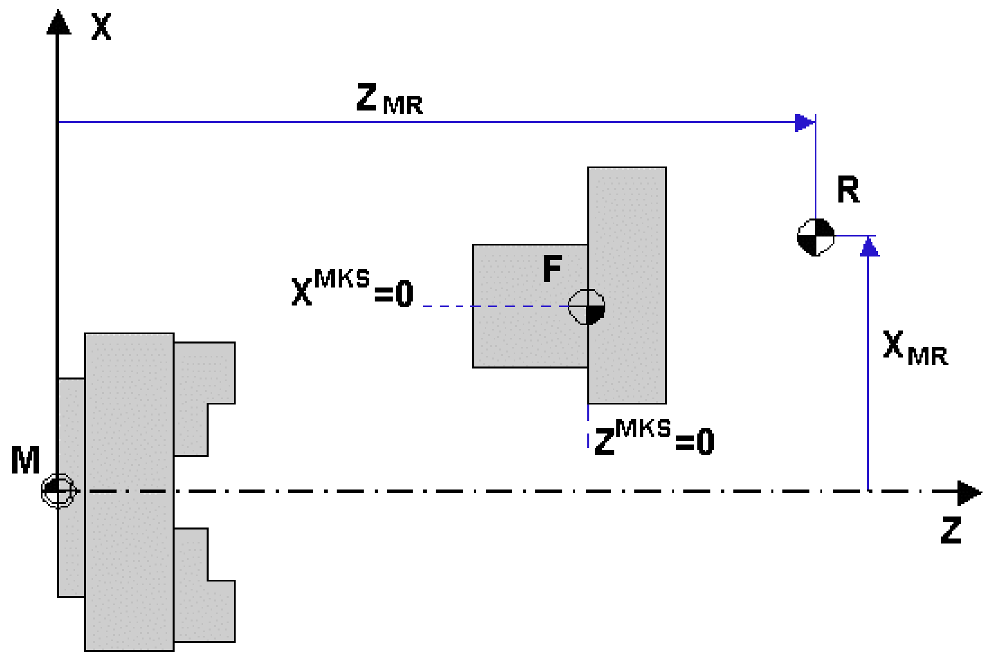


Rys. 24. Programowanie w wyizolowanym układzie przedmiotu (a) oraz realizacja programu na konkretnej obrabiarce z zadaniem wartości przesunięć punktów zerowych i rejestrów narzędziowych (b)

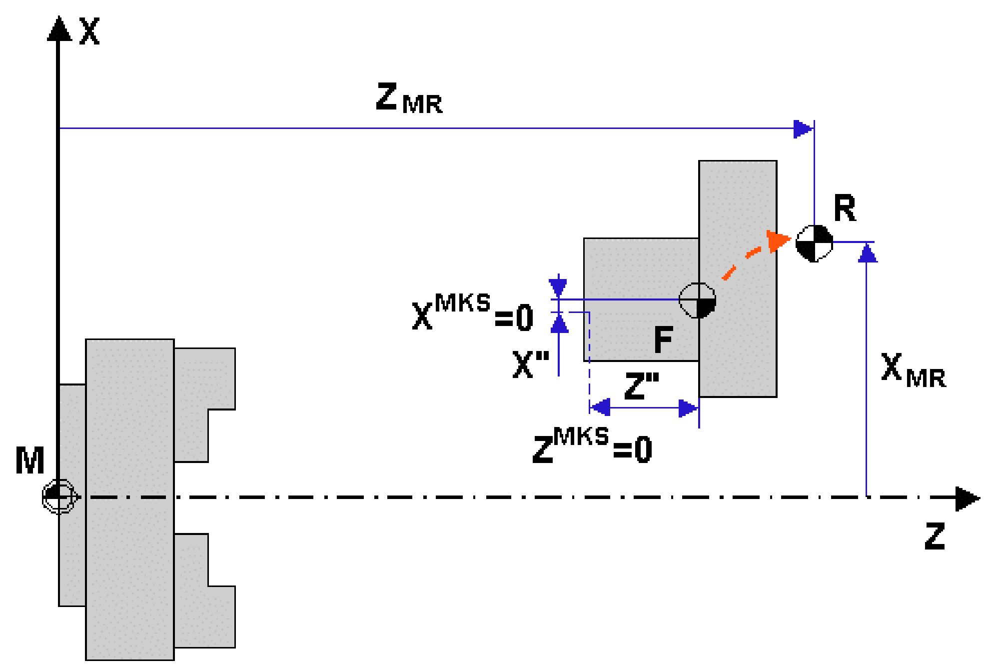
1.6. Najazd na punkt referencyjny

Jak wspomniano wcześniej, układy pomiarowe obrabiarek najczęściej są układami inkrementalnymi (przyrostowymi), nie pozwalającymi wprost dokonywać pomiaru w absolutnym układzie współrzędnych. Dzięki zastosowaniu stałego (bazowego, referencyjnego) punktu dla każdej osi SN możliwe jest zbudowanie absolutnego układu współrzędnych, o czym już wspomniano wcześniej. Warunkiem jest wykonanie po każdym włączeniu obrabiarki najazdu w osiach na te właśnie punkty – nazywane jest to najazdem na punkty referencyjne, wspomagane istnieniem specjalnego trybu pracy układu sterowania CNC. Ideę tego postępowania

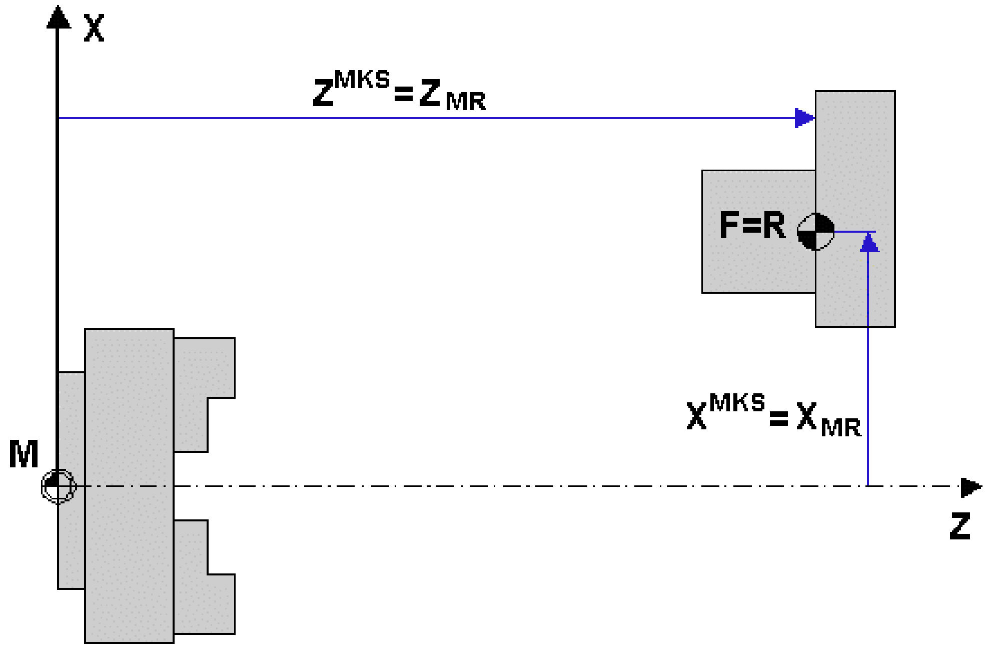
w odniesieniu do zdefiniowanych układów współrzędnych przedstawiono poniżej (Rys. 25, Rys. 26, Rys. 27).



Rys. 25. Stan obrabiarki CNC po włączeniu (wyzerowanie liczników), współrzędne maszynowe równe zero ($Z^{MKS} = 0, X^{MKS} = 0$)



Rys. 26. Najazd na punkt referencyjny – współrzędne wskazywane Z”, X” odmierzane od położenia wyjściowego zespołów ruchomych obrabiarki (innego przy każdym włączeniu obrabiarki)

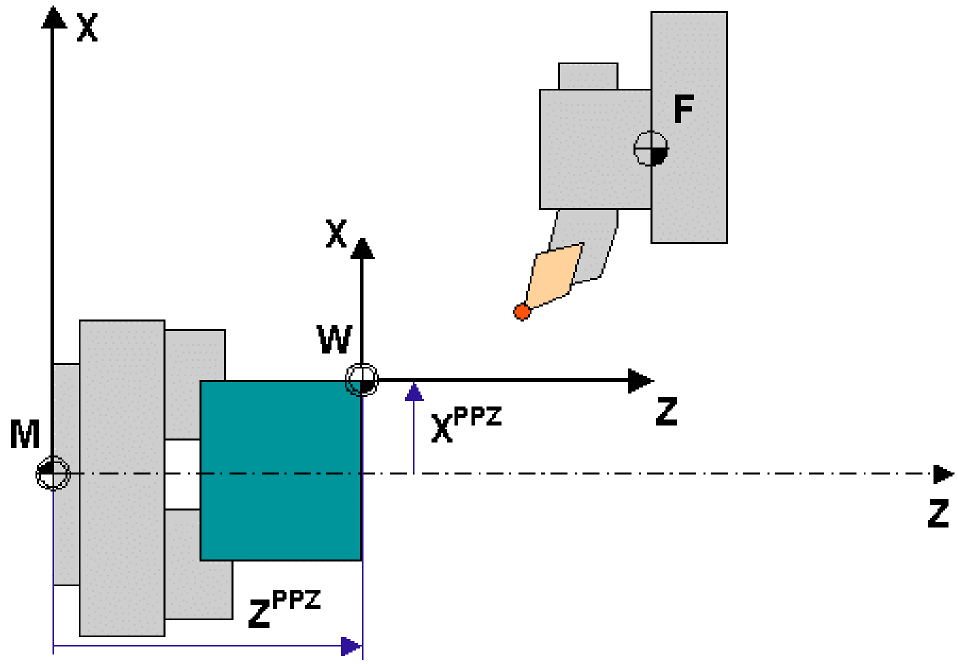


Rys. 27. Ustawienie początku układu MKS (BKS) po najeździe na punkt referencyjny – współrzędne maszynowe przyjmują wartości współrzędnych punktu referencyjnego (pobranych z danych maszynowych)

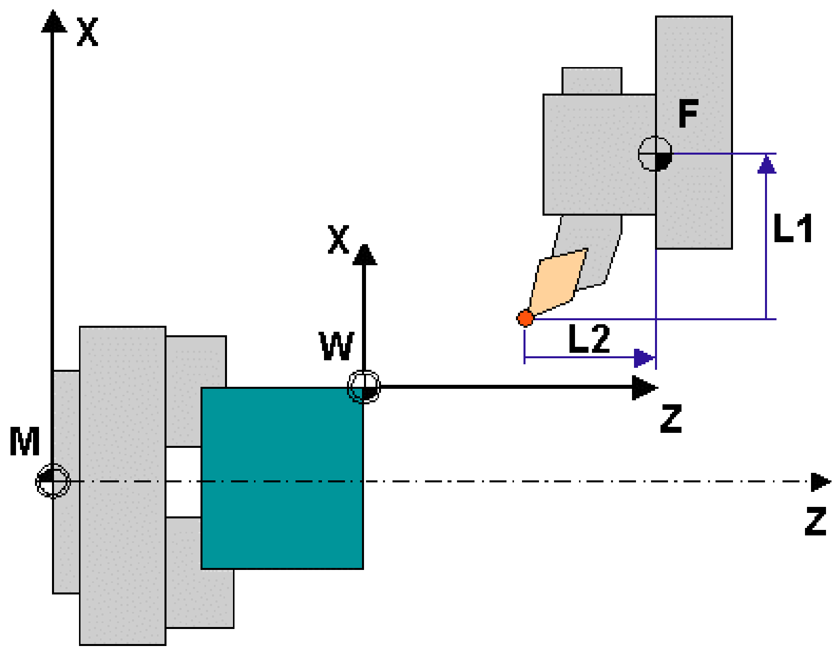
1.7. Zależności pomiędzy współrzędnymi

Zadając współrzędne w układzie przedmiotu układ sterowania musi wyrazić je w układzie maszynowym, aby odpowiednio sterować napędami. Jednocześnie podczas wykonywania programu sterującego możliwe jest wyrażanie aktualnego położenia obrabiarki zarówno we współrzędnych maszynowych, jak i przedmiotowych. Układ CNC musi zatem posiadać informacje o wzajemnych relacjach pomiędzy różnymi układami współrzędnych aby wspomniany proces zachodził w sposób prawidłowy. **Przyjmując układ współrzędnych przedmiotu** można to uczynić w sposób zupełnie dowolny (najczęściej odnoszący się do wymiarowania przedmiotu obrabianego) **należy podać i wprowadzić do układu sterowania** następujące wielkości (por. Rys. 23):

1. Zmianę położenia punktu zerowego (**M** przesunięty na **W**) – **przesunięcia punktu zerowego** (ogólnie X^{PPZ}, Y^{PPZ}, Z^{PPZ}), wprowadzane z pulpitu układu sterowania do rejestrów przesunięć punktów zerowych (Rys. 28).
2. Zmianę punktu kodowego (**F** przesunięty na **P**) – korektory długości narzędzia (ogólnie $L1$, $L2$, $L3$), wprowadzane z pulpitu układu sterowania do rejestrów korektorów narzędziowych (Rys. 29).



Rys. 28. Przesunięcia punktów zerowych (PPZ)

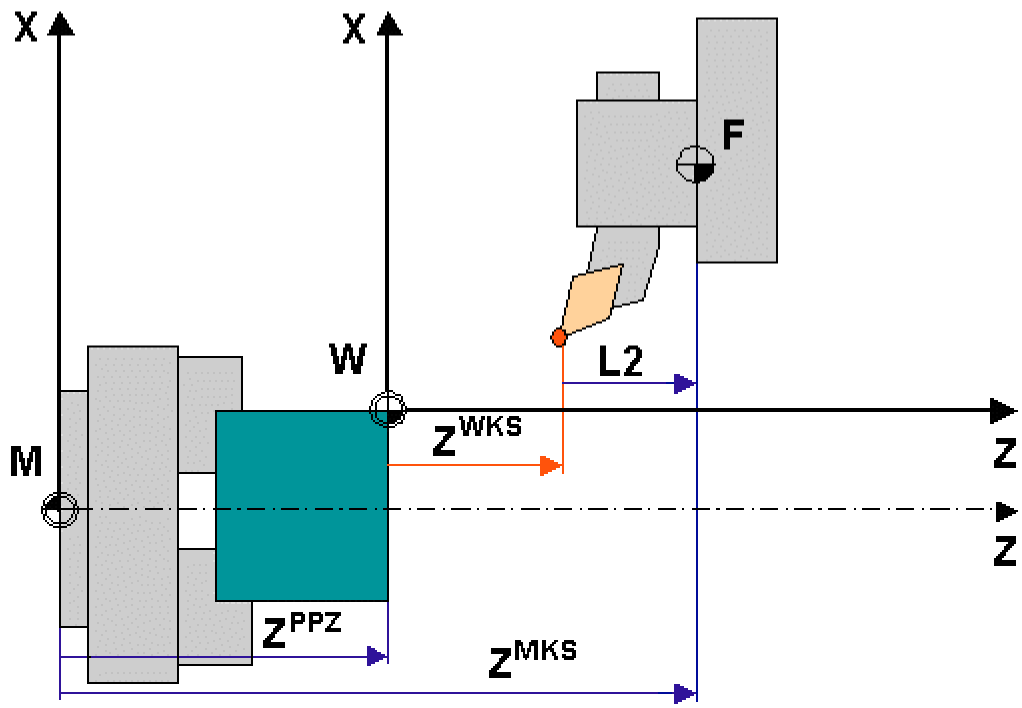


Rys. 29. Wymiary narzędzi (długości korekcyjne)

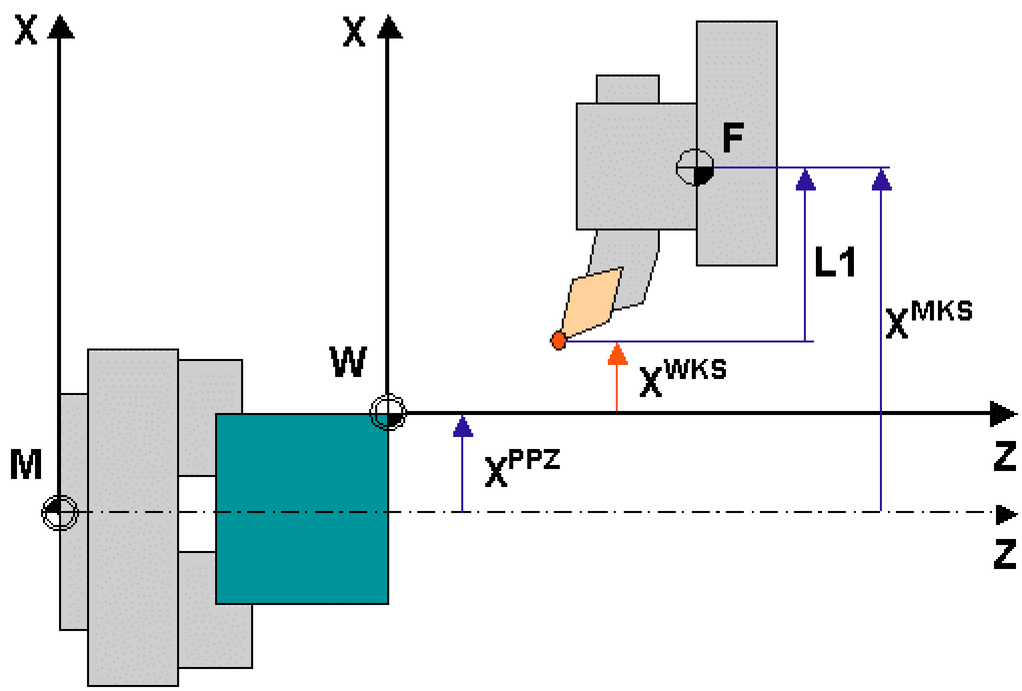
Czynności definiowania rejestrów narzędziowych i rejestrów przesunięć punktów zerowych mają podstawowe znaczenie dla poprawności przebiegu obróbki w trybie automatycznym (sterowanym programem) i należą do najważniejszych czynności przygotowawczych, poprzedzających pracę w trybie automatycznym. Dlatego często są wspomagane specjalnie do tego celu przeznaczonymi funkcjami układu sterowania CNC.

Po ustaleniu tych wielkości układ sterowania jest już w stanie dokonywać przeliczeń współrzędnych położenia osi SN w układach MKS (BKS) i WKS,

uniezależniając tym samym współrzędne toru ruchu narzędzi, zapisane w programie od jednej, konkretnej obrabiarki – dzięki temu program jest uniwersalny. Zależności między tymi układami na przykładzie tokarki przedstawiono na Rys. 30 i Rys. 31 (tylko przy uwzględnieniu przesunięć, bez dodatkowych transformacji). Podobne zależności obowiązują dla innych rodzajów maszyn SN.



Rys. 30. Zależności między współrzędnymi w osi Z tokarki



Rys. 31. Zależności między współrzędnymi w osi X tokarki

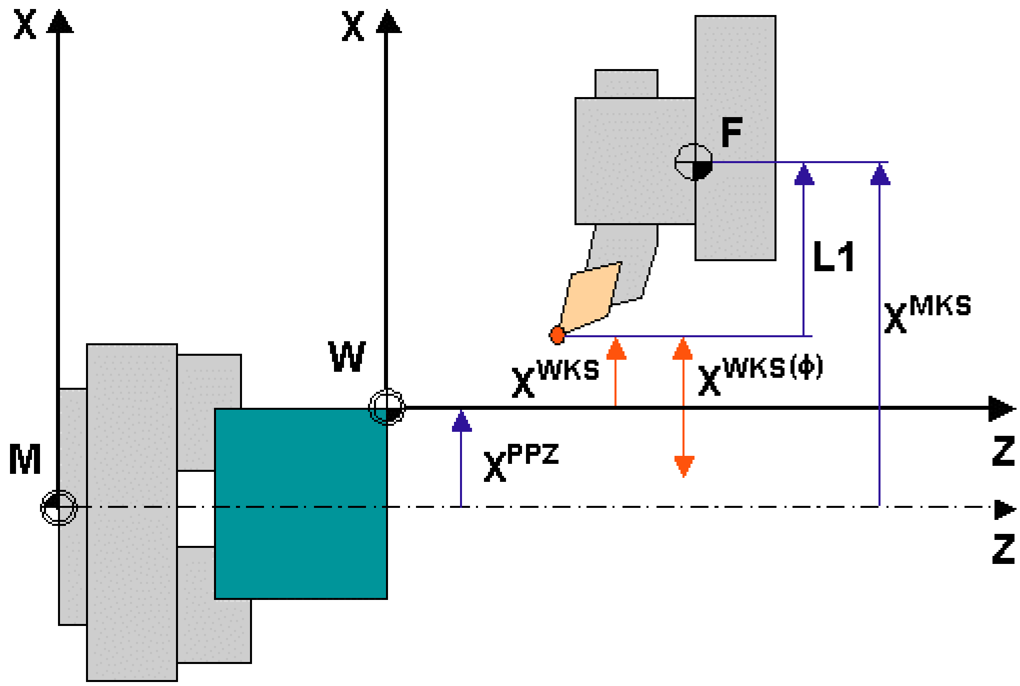
Wtedy możemy wyprowadzić następujące zależności pomiędzy współrzędnymi maszynowymi i przedmiotowymi (dla tokarki):

$$\begin{aligned} Z^{MKS} &= Z^{WKS} + Z^{PPZ} + L2 \\ X^{MKS} &= X^{WKS} + X^{PPZ} + L1 \end{aligned} \tag{1}$$

$$\begin{aligned} Z^{WKS} &= Z^{MKS} - Z^{PPZ} - L2 \\ X^{WKS} &= X^{MKS} - X^{PPZ} - L1 \end{aligned} \tag{2}$$

Wartości rejestrów przesunięć punktów zerowych oraz korektorów narzędziowych są wielkościami wektorowymi, stąd należy pamiętać o odpowiednim znaku przy ich wartościach. Podobne wyrażenia można wyprowadzić na frezarek.

Ponieważ dla obróbki tokarskiej większość wymiarów w osi X wyrażana jest na średnicy, dlatego możliwe jest zadawanie wartości X^{WKS} średnicowo (Rys. 32).



Rys. 32. Współrzędne promieniowe i średnicowe w osi X tokarki

Wówczas:

$$X^{WKS} = \frac{X^{WKS(\phi)}}{2} \tag{3}$$

co prowadzi do następujących zależności:

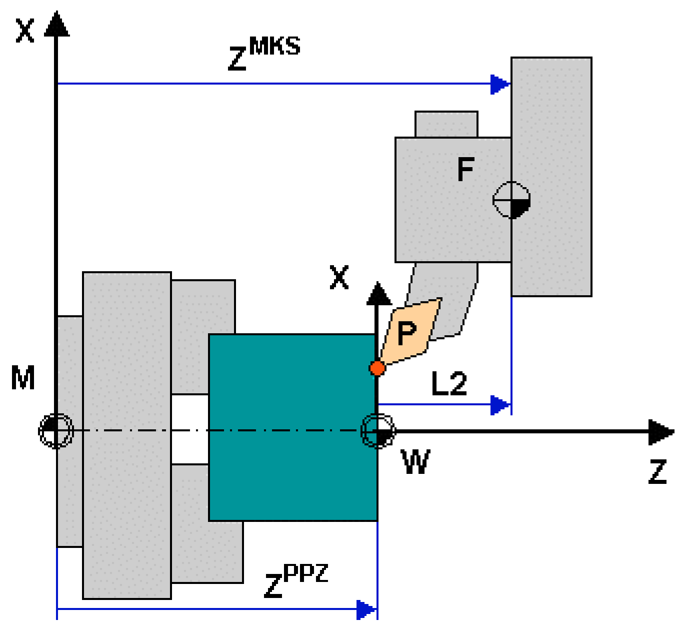
$$\begin{aligned} Z^{MKS} &= Z^{WKS} + Z^{PPZ} + L2 \\ X^{MKS} &= \frac{X^{WKS(\phi)}}{2} + X^{PPZ} + L1 \end{aligned} \tag{4}$$

$$Z^{WKS} = Z^{MKS} - Z^{PPZ} - L2$$
$$X^{WKS(\phi)} = 2 \cdot \left(X^{MKS} - X^{PPZ} - L1\right) \tag{5}$$

1.8. Wyznaczanie wartości rejestrów narzędziowych i rejestrów PPZ

Dla prawidłowego funkcjonowania obrabiarki CNC w trybie automatycznym oprócz programu sterującego konieczne jest **ustalenie poprawnych wartości korektorów narzędziowych i przesunięć punktów zerowych**. Sposób wyznaczenia tych wartości w dużej mierze zależy od rodzaju obrabiarki, jej wyposażenia, dostępności specjalnych urządzeń do pomiaru i nastawiania narzędzi, rodzaju narzędzi itp. Niezależnie od tego można wskazać na pewne ogólne zależności, na podstawie których można opracować własne metody wyznaczania wartości korektorów narzędziowych i przesunięć punktów zerowych.

Rozpatrzmy pewne szczególne położenia narzędzia względem przedmiotu obrabianego, dla którego znana jest wartość co najmniej jednej współrzędnej w układzie przedmiotu (WKS) – np. przy zetknięciu noża z materiałem, na czole którego chcemy umieścić punkt zerowy **W**; wtedy $Z^{WKS} = 0$ (Rys. 33).



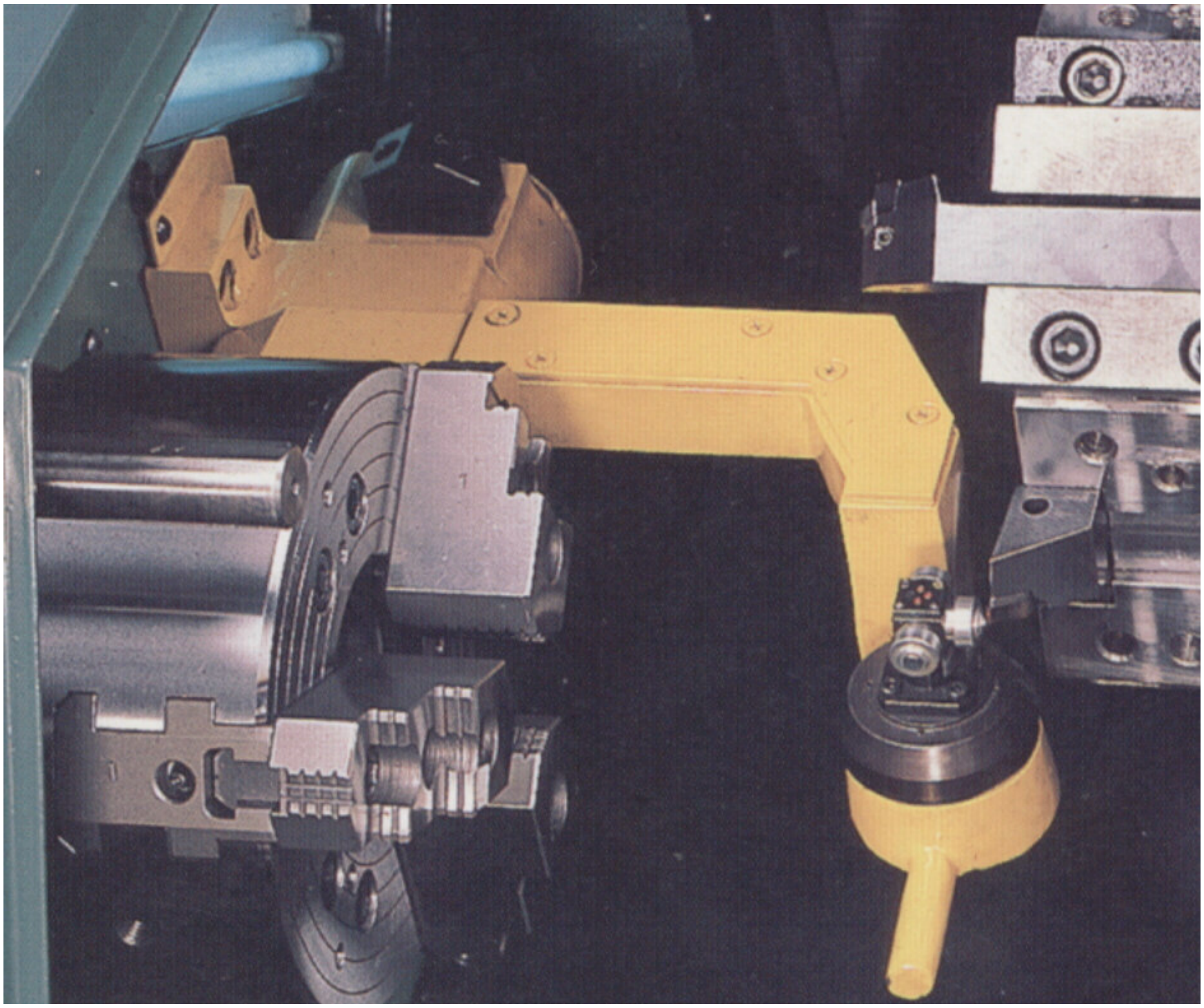
Rys. 33. Zasada wyznaczania wartości przesunięć punktów zerowych i korektorów narzędziowych

Dla tego położenia znane jest również położenie punktu kodowego **F** – współrzędne w układzie maszynowym MKS (Z^{MKS}). W przedstawionych powyżej zależnościach na współrzędne w układzie maszynowym i przedmiotu pozostają jeszcze po dwie wielkości niewiadome – przesunięcia punktu zerowego (Z^{PPZ}) i korektory narzędziowe ($L2$). Musi być zatem znana jeszcze jedna wartość, druga może zostać wyliczona. Często dokonuje się pomiarów narzędzia (wyznaczając jednocześnie korektory narzędziowe), stosując specjalne urządzenia pomiarowo-

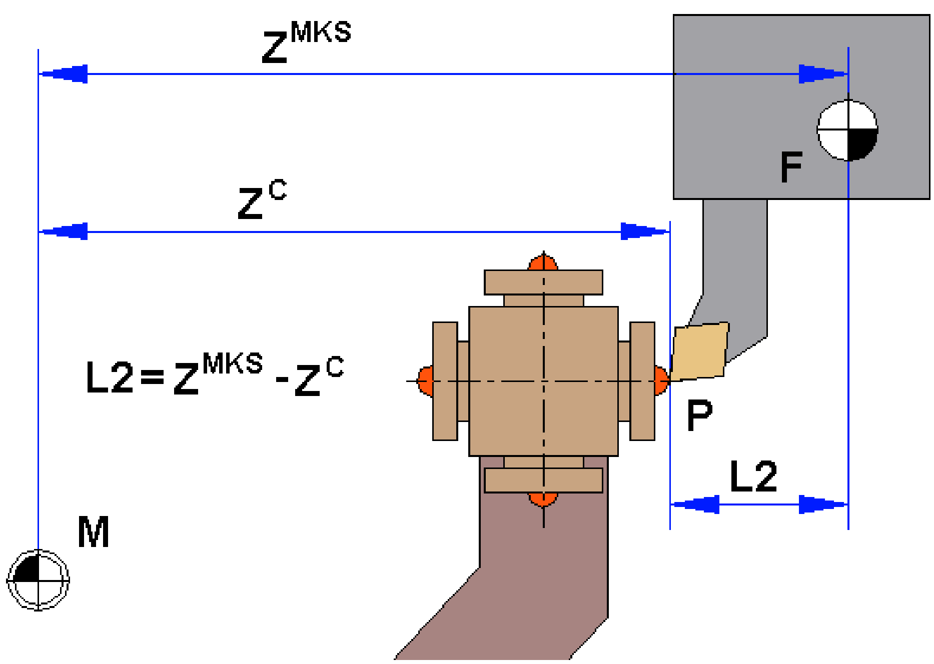
ustawcze (Rys. 34) lub wprost na obrabiarce, wykorzystując jej układ pomiarowy (obrabiarka musi być wyposażona w przyrządy stykowe o znanym położeniu w przestrzeni roboczej obrabiarki – Rys. 35, Rys. 36).



Rys. 34. Urządzenie optyczne do nastawiania i pomiarów narzędzi poza obrabiarką (f. Trimos)



Rys. 35. Pomiar narzędzia na obrabiarce za pomocą uchylnego wysięgnika z końcówkami stykowymi (f. Goodway)



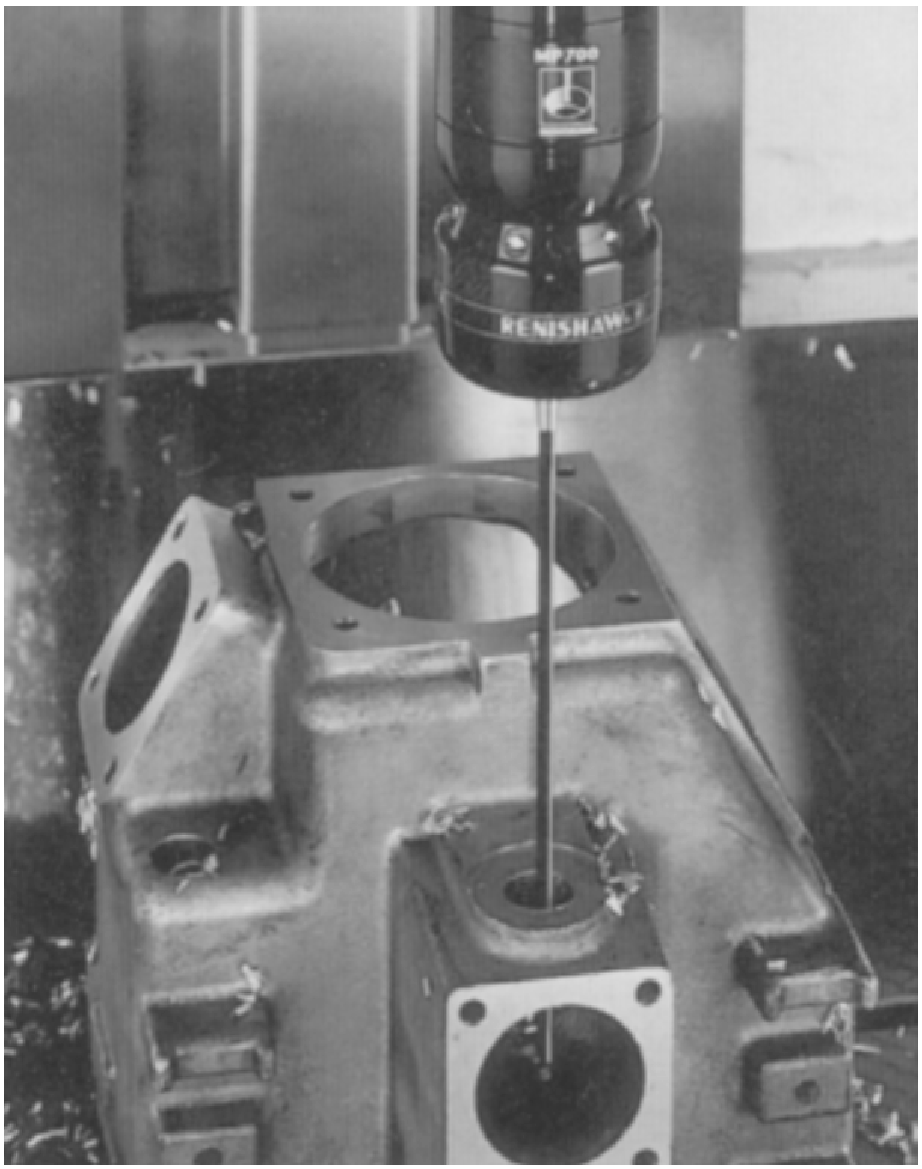
Rys. 36. Zasada pomiaru narzędzia na obrabiarce

Na tej podstawie wylicza się przesunięcia punktu zerowego (Rys. 33):

$$
\begin{aligned}
Z^{PPZ} &= Z^{MKS} - Z^{WKS} - L2 \\
X^{PPZ} &= X^{MKS} - \frac{X^{WKS(\phi)}}{2} - L1
\end{aligned}
\qquad (6)
$$

Wykorzystując układ pomiarowy obrabiarki i wyposażając je w sondy pomiarowe (podobne do stosowanych na współrzędnościowych maszynach pomiarowych – Rys. 37) można również dokonać pomiarów przedmiotu obrabianego, a tym samym położeń punktów zerowych układu przedmiotu. Wtedy należy wyliczyć korektory narzędziowe (Rys. 33):

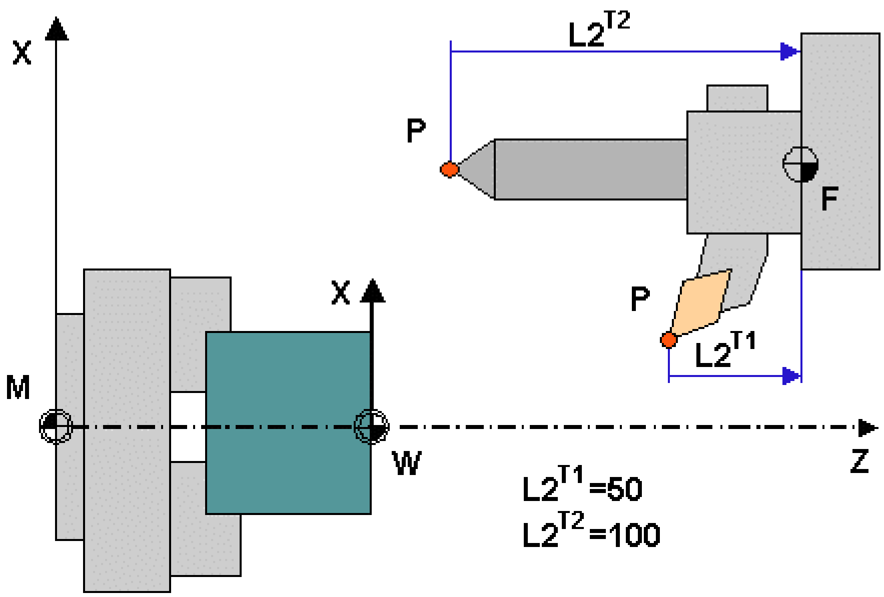
$$
\begin{aligned}
L2 &= Z^{MKS} - Z^{WKS} - Z^{PPZ} \\
L1 &= X^{MKS} - \frac{X^{WKS(\phi)}}{2} - X^{PPZ}
\end{aligned}
\qquad (7)
$$



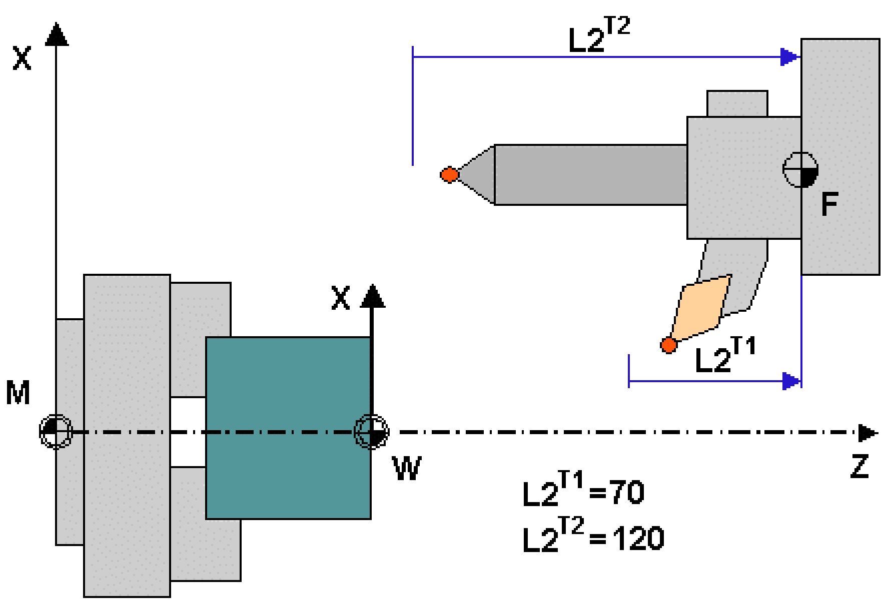
Rys. 37. Pomiar przedmiotu obrabianego na obrabiarce za pomocą głowicy stykowej (f. Renishaw)

W układach sterowania na ogół są funkcje półautomatycznego wyznaczania tych wartości, dlatego nie trzeba ręcznie wyliczać podanych wartości. Dużym ułatwieniem przy wyznaczaniu powyższych danych jest fakt, iż **nie muszą one zawsze odpowiadać wartościom rzeczywistym**; ważniejsze jest podanie np. w korektorach narzędziowych **wartości odpowiadającym różnicom wymiarów narzędzi** niż ich rzeczywistym wymiarom.

Rozpatrzmy zatem dwa przykłady, w których użyto dwa narzędzia, przy czym w pierwszym znane są rzeczywiste długości narzędzi (Rys. 38), w drugim natomiast są one większe, ale zachowana została ich różnica (Rys. 39).

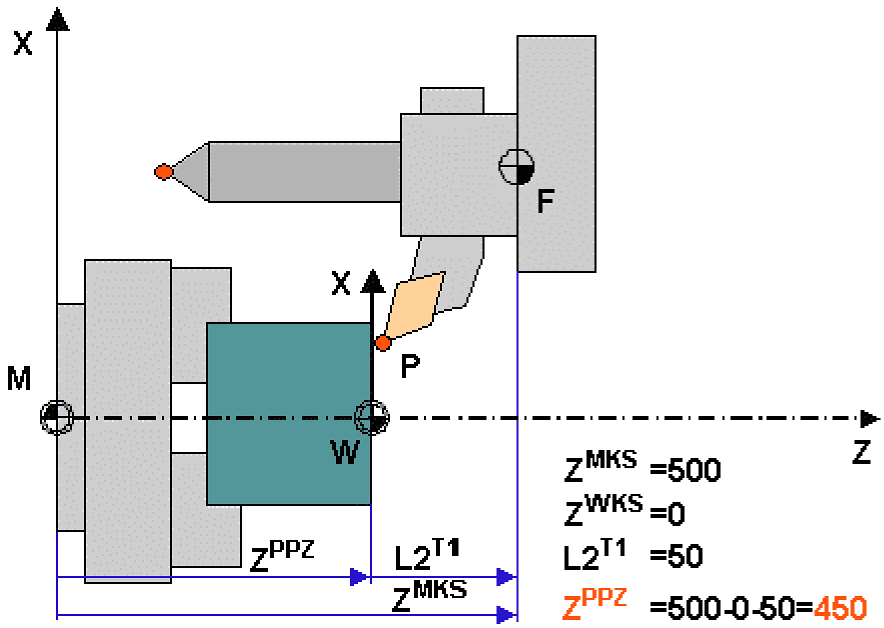


Rys. 38. Przykład I – rzeczywiste długości narzędzi

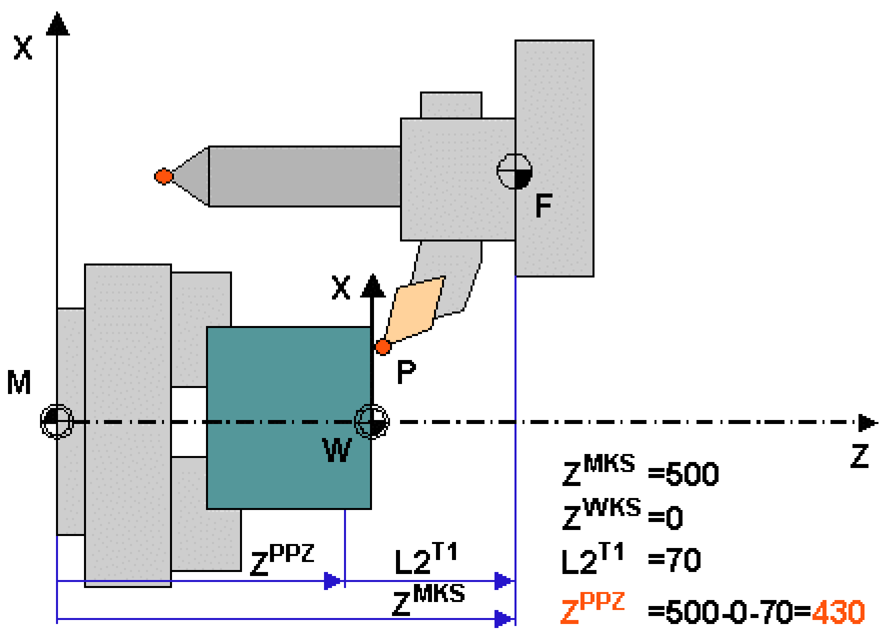


Rys. 39. Przykład II – zwiększone długości narzędzi

Narzędzia T1 w obu przykładach użyto do wyznaczenia wartości przesunięcia punktu zerowego, w wyniku czego uzyskano dwie różne wartości tego przesunięcia (Rys. 40 i Rys. 41).

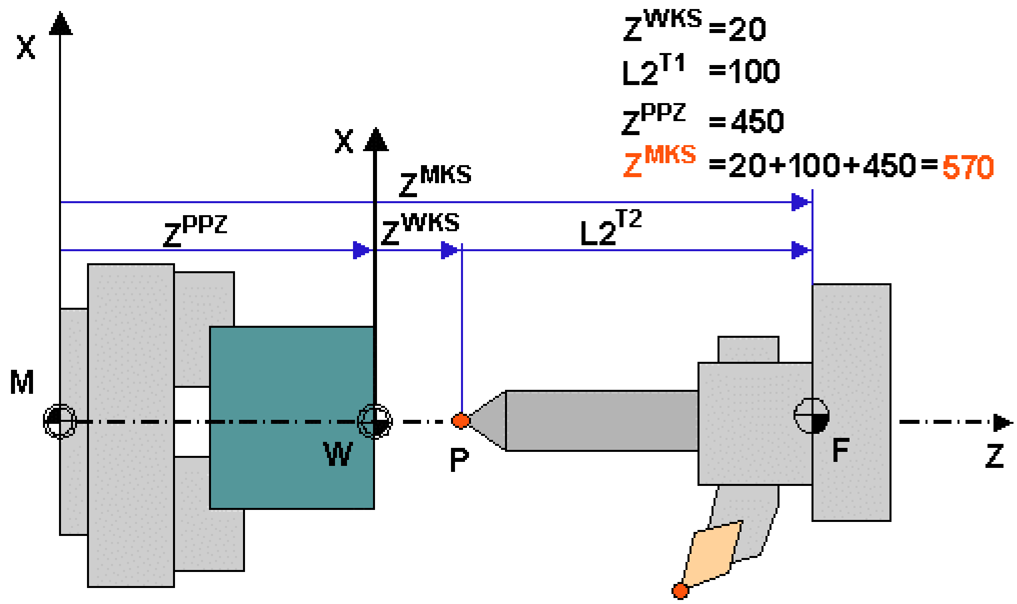


Rys. 40. Wyznaczanie przesunięcia punktu zerowego dla przykładu I

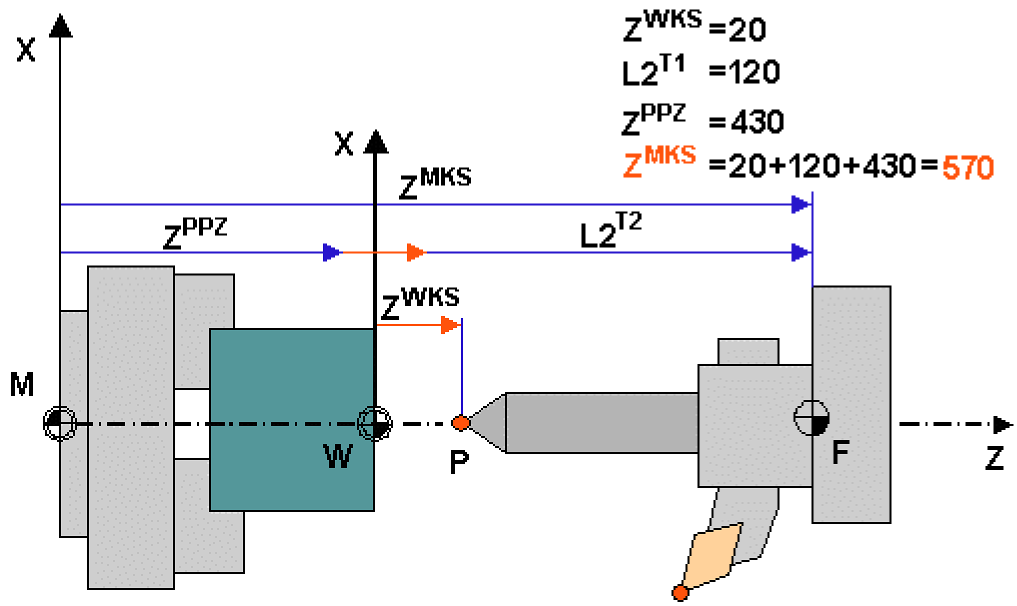


Rys. 41. Wyznaczanie przesunięcia punktu zerowego dla przykładu II

Jeżeli teraz w układzie przedmiotu (WKS) zadamy położenia narzędzia T2 równe $Z^{WKS} = 100$ to dla obu przykładów uzyskamy następujące położenia punktu kodowego **F** w układzie maszynowym (Rys. 42 i Rys. 43).



Rys. 42. Ustalanie położenia narzędzia T2 w układzie WKS dla przykładu I



Rys. 43. Ustalanie położenia narzędzia T2 w układzie WKS dla przykładu II

Jak łatwo zauważyć, w obu przykładach uzyskano **te same wartości współrzędnej** Z^{MKS} (a tym samym to samo położenie suportu narzędziowego), co przy identycznym położeniu przedmiotu obrabianego, identycznych długościach narzędzi i identycznym położeniu układu przedmiotu (WKS) jest zrozumiałe, choć zostało uzyskane przy różnych wartościach korektorów narzędziowych i przesunięć punktów zerowych. Wniosek jest zatem następujący (co jest też widoczne na powyższych rysunkach): **niedokładność pomiaru długości narzędzi została skompensowana położeniem punktu zerowego W.** Powyżej przedstawione właściwości w połączeniu z zaletami przyrostowego układu pomiarowego są często wykorzystywane w praktyce, eliminując konieczność stosowania dodatkowego (a także drogiego) oprzyrządowania ustawczo-pomiarowego.

2. STRUKTURA PROGRAMU STERUJĄCEGO

2.1. Wprowadzenie

Operacja obróbki jest ciągiem ruchów wykonywanych przez narzędzie względem przedmiotu obrabianego, uzupełnionych czynnościami pomocniczymi. Program sterujący jest też ciągiem instrukcji kodujących te ruchy poprzez zapis współrzędnych, uzupełnionych instrukcjami pomocniczymi. Zapis elementarnego ruchu (czynności) jest nazywany **blokiem** (czasem również **zdaniem**), przy czym blok może również zawierać inne zapisy, potrzebne do wykonania ruchu (np. wymiana narzędzia czy ustalenie parametrów obróbki). Program sterujący (zwany też programem głównym) jest zatem ciągiem **bloków**, najczęściej zapisywanych w edytorze w oddzielnych liniach (dłuższe bloki mogą zajmować więcej linii stąd **linia** nie jest równoznaczna pojęciu **bloku**):

```
Blok_1
Blok_2
....
Blok_n
```

Blok jest przez układ sterowania traktowany jako pewna **całość**, **w całości czytanym z programu sterującego, analizowanym i wykonywanym** (w praktyce układ sterowania czyta jednocześnie kilka bloków co wynika m.in. z konieczności zachowania ciągłości obliczeń ciągów konturowych, kompensacji promienia narzędzia itp.). O kolejności wykonania bloków decyduje ich kolejność w programie sterującym, o ile nie są stosowane zaawansowane techniki programowania, np. skoki czy pętle (co zostanie omówione w następnych rozdziałach).

Specjalną, ustaloną dla danego układu sterowania, postać mają pierwszy i ostatni blok programu. Pierwszy, nazywany **nagłówkiem programu**, pełni podwójną rolę:

- ➔ zawiera informację o typie danych przechowywanych w danym pliku (oprócz programu może to być także podprogram, zawartość rejestrów narzędziowych, rejestrów przesunięć punktów zerowych, R-parametrów, danych maszynowych itp.) – odpowiednik rozszerzenia pliku w zwykłym komputerze;
- ➔ w przypadku programów (i podprogramów) zawiera informację o jego nazwie – odpowiednik nazwy pliku w zwykłym komputerze.

W układzie sterowania nagłówek programu ma postać:

%_N_PROGRAM1_MPF

a podprogramu:

%_N_PODPROGRAM1_SPF

MPF (ang. *Main Program File*) jest rozszerzeniem dla programu, a SPF (ang. *SubProgram File*) – podprogramu. Nazwy *PROGRAM1* i *PODPROGRAM1* są nazwami przykładowymi. W każdym układzie sterowania nazwom są stawiane inne wymagania – dopuszczalne znaki, długość itp. Poprawna forma nagłówka ma znaczenie przede wszystkim przy transmisji do układu sterowania – wtedy w sterowniku CNC tworzony jest plik **o nazwie pobranej z nagłówka**, a nie o nazwie pliku dyskowego, **w którym na komputerze był on przechowywany**. Podobnie przy

transmisji z układu sterowania – sterownik automatycznie dodaje nagłówek do transmitowanych danych. Przy przechowywaniu plików programów i podprogramów na zwykłym komputerze powinno się zachować zgodność nagłówków z nazwą pliku w którym program/podprogram się znajduje (choć nie jest to obowiązkowe). Np. nazwie programu „_N_PROGRAM1_MPF" powinien odpowiadać plik „PROGRAM1.MPF".

Drugim ważnym blokiem jest blok ostatni, zawierający znak końca programu, podprogramu bądź innej struktury danych (nie mylić **znak końca programu** ze **znakiem końca pliku** – EOF, ang. *End Of File*). Informuje on układ sterowania o zakończeniu wykonywania programu. W programach o rozgałęzionej strukturze realizacji bloków może się znajdować kilka znaków końca programu (niekoniecznie w ostatnim bloku). W układzie sterowania znak końca programu to **M30** lub **M2**, podprogramu – **M17**. Dwa rodzaje zakończenia programu wynikają z przesłanek historycznych, obecnie są to dwa równoważne zapisy, przy czym autor sugeruje stosowanie znaku **M30** ponieważ stanowi on zakończenie również innych danych. Zatem najprostszy program lub podprogram posiada następującą strukturę, gdzie nagłówek i zakończenie stanowią jedyne obowiązkowe elementy, pomiędzy którymi wprowadza się właściwy program (podprogram):

```
%_N_PROGRAM1_MPF
M30

%_N_PODPROGRAM1_SPF
M17
```

Blok też jest strukturą złożoną, składającą się ze **słów**, które pozwalają na wywoływanie elementarnych funkcji układu sterowania:

*Słowo*_1 *Słowo*_2 .... *Słowo_m* **LF**

Kolejność słów w bloku nie ma znaczenia (z pewnymi wyjątkami), ponieważ analiza treści bloku jest realizowana w stosunku do całego bloku, a nie jego pojedynczych elementów. Ostatnim, obowiązkowym elementem bloku jest słowo **końca bloku** (LF, ang. *Line Feed*). Ma on decydujące znaczenie dla układu sterowania, który zawsze czyta program z pamięci sekwencyjnie od początku bloku do końca bloku, niezależnie czy na ekranie monitora blok zajmuje jedną, dwie lub więcej linii – zależy to od wielkości ekranu i ustawień wyświetlania programu (**dlatego nie należy mylić linii programu z blokiem programu**). Graficznie znak końca bloku może mieć różną postać, może też być wyłączone jego wyświetlanie (nie jest to zalecane).

Słowo z kolei składa się na ogół z dwóch elementów: **Adresu** i **Wartości**. Adres należy rozumieć jako nazwę elementarnej funkcji układu sterowania, natomiast wartość – argumenty tej funkcji (stąd istnieją też słowa składające się tylko z adresu – funkcje bezparametryczne). Zapis słowa może być różny, w zależności od rodzaju adresu. Można tu wyróżnić:

1. Słowa proste, gdzie adresy składają się z jednej, dużej litery alfabetu łacińskiego; wtedy wartość pisze bezpośrednio po adresie, np. **M30**.
2. Słowa złożone, gdzie adresy składają się z kilku dużych liter alfabetu łacińskiego; wtedy wartość pisze się po znaku „=", np. **AP=30**.

3. Słowa rozszerzone, odnoszące się np. do wrzeciona o danym numerze (wtedy numer ten jest rozszerzeniem słowa); wtedy bezpośrednio po adresie występuje rozszerzenie, po nim znak „=”, a po nim wartość adresu, np. **S2=300** (słowo odnosi się do wrzeciona nr 2).
4. Słowa z wartością pośrednią (np. za pomocą tzw. R-parametrów) wymagają po adresie znaku „=”, np. **X=R20**. Szerzej ten temat jest omawiany w dalszej części.

Zapis wartości słowa podlega następującym zasadom:

1. Niektóre adresy wymagają wartości całkowitej lub naturalnej wartości (np. określające numer narzędzia), pozostałe mogą mieć wartość rzeczywistą.
2. Separatorem dziesiętnym jest znak kropki „.”, np. **X23.6**.
3. Wartości dodatnie na ogół nie wymagają podania znaku „+”, choć podanie go nie jest błędem, np. **X+23.6** (nie należy rozumieć tego jako zapis operacji dodawania).
4. Wartości ujemne wymagają wprowadzenia znaku „-”, np. **X-23.6**.
5. Precyzja podawania części ułamkowej jest zależna od układu sterowania, na ogół wartości adresów można podawać z dokładnością do 3 lub 4 miejsc po przecinku, np. **X.23.678**.
6. Jeżeli część całkowita jest równa 0 to można ją opuścić, np. **F.2** oznacza dokładnie to samo co **F0.2**.
7. Można zaznaczyć, iż wartość jest typu rzeczywistego, choć część ułamkowa jest równa zero przez pominięcie tej części z pozostawieniem kropki dziesiętnej, np. **X23.** oznacza dokładnie to samo co **X23.0**.

Zapis słowa nie może zawierać spacji (i innych separatorów), słowa mogą (ale nie muszą) być oddzielone od siebie znakiem spacji (program jest wtedy bardziej czytelny).

2.2. Podstawowe adresy

Zestaw podstawowych adresów obejmuje przede wszystkim adresy proste. Najważniejsze z nich dla układu Sinumerik 840D, to:

A, **B**, **C**	– wartości współrzędnych w osiach obrotowych A, B i C
D	– numer rejestru narzędziowego
F	– programowanie posuwu/czasu postoju
G	– funkcje przygotowawcze
H	– funkcje dodatkowe
I, **J**, **K**	– parametry interpolacji w osiach odpowiednio X, Y i Z
L	– wywołanie podprogramu
M	– funkcje pomocnicze (maszynowe)
N	– numer bloku
P	– krotność wywołania podprogramu
R	– programowanie z wykorzystaniem R-parametrów
S	– programowanie obrotów wrzeciona/prędkości skrawania/czasu postoju
T	– ustawienie narzędzia w magazynie narzędziowym
X, **Y**, **Z**	– wartości współrzędnych w osiach odpowiednio X, Y, Z.

Większość adresów zostanie szczegółowo omówiona w dalszych rozdziałach, jedynym adresem, który nie jest objęty dokumentacją producenta układu sterowania,

jest adres H (niem. *Hilfsfunktion*). Jest on zarezerwowany do użytku producentów obrabiarek (często łącznie z adresem M), którzy mogą z niego korzystać przy oprogramowaniu niestandardowych funkcji swoich produktów. Zatem ich opisu należy poszukiwać w dokumentacji techniczno-ruchowej konkretnej obrabiarki.

Przy programowaniu obowiązuje zasada, iż **adres może wystąpić dokładnie jeden raz w bloku** (nie dotyczy jedynie adresów G i M). W przeciwnej sytuacji generowany jest błąd.

2.3. Numer bloku N

Adres N (ang. *block Number*) jest jedynym słowem, który ma stałe miejsce w bloku – zawsze musi być pierwszym adresem w bloku. Numer bloku nie wywołuje żadnej czynności obrabiarki, jest tylko pewną etykietą (opisem) bloku, w którym się znajduje, dlatego we większości układów sterowania nie jest obowiązkowy. Mimo to jest zalecane stosowanie numerowania bloków. Jest to podyktowane kilkoma okolicznościami:

1. Podczas edycji obszernych programów numer bloku informuje operatora/programistę czy jest na początku, końcu czy w środku programu.
2. Numer bloku pozwala szybko wyszukać ten blok w edytorze programów.
3. Przy wystąpieniu błędu na ogół układ sterowania podaje również numer bloku, w którym ten błąd wystąpił – szybsza diagnostyka błędów.
4. Istnieje funkcja rozpoczynania programu nie od początku, ale od wyszukanego w programie elementu, najczęściej jest nim właśnie numer bloku.
5. Możliwe jest wpływanie na wykonanie bloku przez uczynienie go blokiem warunkowym, tzn. poprzez poprzedzenie adresu N znakiem „/". Blok warunkowy jest wykonywany, jeżeli z poziomu pulpitu układu sterowania jest nieaktywna funkcja SKIP BLOCK. Jeżeli funkcja ta jest aktywna – blok warunkowy nie jest wykonywany. Daje to prostą metodę na realizację programu wielowariantowego.

Numerowanie bloków może odbywać się na dowolnych, określonych przez programistę zasadach. Jednak najczęściej numeruje się bloki rosnąco, co określoną wartość, np. co 5 czy 10. Zawsze istnieje możliwość przenumerowania bloków programu, o ile dodano lub usunięto z programu jakieś bloki, co zakłóciło istniejącą numerację.

Przykład programu z numerami bloków:

```
N05 G54 G71
/N10 T1 D1                    blok warunkowy
N15 X90 Y20
```

2.4. Funkcje przygotowawcze G

Adres G (ang. *Preparatory function*) to jeden z najważniejszych adresów. Choć funkcje przygotowawcze nie wywołują żadnych czynności obrabiarki to ich zadaniem jest **interpretowanie znaczenia innych adresów**. Np. sam zapis **X10**, odnoszący się do współrzędnej w osi X nie jest jednoznaczny, nie wiadomo dokładnie co powinien spowodować. Wynika to dopiero z użytych funkcji przygotowawczych. Wśród funkcji przygotowawczych są również takie, które mają inny adres niż G.

Funkcje przygotowawcze mają specyficzne działanie i dlatego też specyficzna jest ich organizacja. Ogólnie adresy używane w układzie sterowania (w tym funkcje przygotowawcze) dzielą się na **dwie grupy**:

1. **Adresy modalne** (**globalne**), obowiązujące w programie aż do ich odwołania – są aktywne w bloku nawet jeżeli w tym bloku nie są wywoływane.
2. **Adresy niemodalne** (**lokalne**), obowiązujące tylko dla bloku w którym zostały wywołane, lub adresu z którym występują – nie ma konieczności ich odwoływania.

Funkcje przygotowawcze (zarówno modalne jak i niemodalne) zostały podzielone na **grupy funkcji** o zbliżonym działaniu, przy czym dla **funkcji modalnych** obowiązują następujące zasady:

1. Tylko **jedna** funkcja z grupy może być **aktywna**.
2. **Wywołanie jednej funkcji** z grupy **automatycznie odwołuje** działanie dotychczas aktywnej funkcji.
3. **W jednym bloku** możliwe jest wywołanie tylko **jednej funkcji danej grupy** – w jednym bloku można co najwyżej użyć tylu funkcji G, ile jest grup funkcji przygotowawczych.
4. **Zawsze jest aktywna** jakaś funkcja danej grupy – w układzie sterowania producent obrabiarki wstępnie aktywuje domyślne funkcje z każdej grupy funkcji modalnych. Nie jest zatem konieczne przywoływanie w programie **domyślnej funkcji danej grupy** – jest ona już aktywna w momencie rozpoczęcia działania programu.

Omawiane w dalszych rozdziałach funkcje przygotowawcze będą zawsze w jednej grupie, przy czym zostanie wskazana funkcja domyślna (za pomocą *). Nie jest to zawsze spełnione, należy zawsze na układzie sterowania sprawdzić listę aktywnych funkcji przygotowawczych – może się różnić od podanej w niniejszym skrypcie.

2.5. Funkcje technologiczne S, F

Podstawowe znaczenie adresu S (ang. *Speed*) odnosi się do programowania prędkości głównego ruchu skrawania, którego zadaniem jest umożliwienie skrawania. Nie ma on natomiast żadnego wpływu na tor ruchu narzędzia i nie jest wymagany przy jego programowaniu. Domyślny sposób określania prędkości odbywa się przez zadanie liczby obrotów wrzeciona głównego (narzędziowego lub przedmiotowego) w jednostce czasu [obr/min]. Inne sposoby programowania tej prędkości wymagają stosowania funkcji przygotowawczych, co omówiono w następnych rozdziałach. Dotyczy to również drugiego znaczenia adresu S – programowanie postoju czasowego.

Drugi z adresów technologicznych – F (ang. *Feed*) – w swoim podstawowym znaczeniu odnosi się do programowania prędkości posuwu. Posuw w znaczący sposób związany jest z kształtowaniem przedmiotu obrabianego i jest w związku z tym wymagany przy programowaniu toru narzędzia. W zależności od rodzaju obrabiarki posuw jest programowany w [mm/obr] – tokarka lub [mm/min] – frezarka. Inne sposoby wyrażania posuwu (oraz programowanie postoju czasowego), podobnie jak prędkość skrawania wymagają stosowania odpowiednich funkcji przygotowawczych, co omówiono w dalszych rozdziałach.

2.6. Funkcje narzędziowe T, D

Adres T (ang. *Tool*) wywołuje **zmianę położenia magazynu narzędziowego**. Zadanie konkretnej wartości (która musi być typu naturalnego) powoduje ustawienie się magazynu narzędziowego w ten sposób, że na jego aktywnej pozycji znajdzie się narzędzie kodowane poprzez zadany numer. Rozumienie aktywnej pozycji zależy od sposobu realizacji przechowywania narzędzi i ich uczestnictwa w obróbce. W obrabiarkach typu tokarka magazyn narzędziowy (np. w postaci obrotowej tarczy narzędziowej) jednocześnie pełni rolę imaka narzędziowego dla narzędzia w trakcie obróbki, co oznacza, że po przywołaniu adresu T narzędzie o podanym numerze jest gotowe do obróbki. We frezarkach narzędzia na ogół są przechowywane w magazynie typu łańcuchowego, tarczowego itp. a przed obróbką za pomocą dodatkowego urządzenia (nazywanego zmieniaczem) przenoszone do wrzeciona narzędziowego. Dla ułatwienia rozróżniania narzędzi przez układ sterowania często wyposaża się je w specjalne wkładki identyfikacyjne (Rys. 44), w których przechowywane są informacje o numerze narzędzia, jego parametrach, czasie pracy itp.



Rys. 44. Wkładki identyfikacyjne dla narzędzi skrawających (f. Sandvik Coromant)

Po przywołaniu adresu T narzędzie o podanym numerze jest **gotowe do wymiany** (jednak nadal znajduje się w magazynie). Do wywołania **zmiany narzędzia** służą inne funkcje, może to być należące do grupy funkcji pomocniczych słowo **M6**.

Adres D (ang. *tool offset number*) jest numerem tzw. **rejestru narzędziowego**. Rejestr narzędziowy (bardziej szczegółowo omówiony w dalszych rozdziałach) to zestaw parametrów opisujących narzędzie. Należą do nich omówione wcześniej wymiary L1 i L2 (dla noży tokarskich). Przywołanie rejestru narzędziowego wiąże się

więc ściśle z definicją układu współrzędnych przedmiotu (WKS) i jest warunkiem poprawnego kształtowania przedmiotu obrabianego.

2.7. Funkcje pomocnicze (maszynowe) M

Funkcje pomocnicze M (ang. *Miscellaneous function*) czasami nazywane funkcjami maszynowymi, w starszych układach sterowania były przeznaczone do bezpośredniej obsługi urządzeń obrabiarki – najczęściej do **sterowania osi dyskretnych**. Część z tych funkcji jest standardowa, większość jednak (w połączeniu także z adresem H) służy do obsługi specyficznych dla danej obrabiarki urządzeń. Stąd dokładnego ich opisu należy poszukiwać w dokumentacji techniczno-ruchowej danej obrabiarki. Istnieje ograniczenie liczby funkcji pomocniczych w jednym bloku. W układzie sterowania Sinumerik 840D jest to max. 5 funkcji.

Do najczęściej stosowanych standardowych funkcji pomocniczych należą:

M0 – bezwarunkowe zatrzymanie wykonania programu;
M1 – warunkowe zatrzymanie wykonania programu.

Zatrzymanie wykonania programu oznacza, że następuje wyłączenie posuwu i obrotów wrzeciona, po czym możliwa jest ingerencja operatora w przestrzeń roboczą obrabiarki (np. w celu wykonania pomiarów). Ponowne uruchomienie programu powoduje wykonywanie bloków po bloku z funkcją zatrzymania. Różnica pomiędzy zatrzymanie warunkowym a bezwarunkowym polega na tym, iż dla bezwarunkowego zatrzymanie wykonania programu jest respektowane zawsze, natomiast dla warunkowego jest zależne od dodatkowej funkcji sterującej (OptM1), ustawianej z pulpitu operatorskiego.

M2 – zakończenie wykonywania programu głównego;
M17 – zakończenie wykonywania podprogramu;
M30 – zakończenie wykonywania programu głównego.

Funkcje M2 lub M30 (o identycznym działaniu) powodują, że zostaje zakończona analiza i wykonywanie bloków programu głównego, nawet jeżeli po bloku z tymi funkcjami są jeszcze jakieś bloki w programie sterującym. Na ogół jednak funkcje te znajdują się w ostatnim bloku programu. Podobne działanie ma funkcja M17.

M3 – włączenie prawych obrotów wrzeciona;
M4 – włączenie lewych obrotów wrzeciona;
M5 – wyłączenie obrotów wrzeciona.

Przed zaprogramowaniem włączenia obrotów należy zadać wartość prędkości obrotowej (patrz adres S). Prawe obroty wrzeciona oznaczają, iż patrząc w kierunku dodatnim osi Z (od tyłu wrzeciennika) wrzeciono obraca się zgodnie z ruchem wskazówek zegara (stąd często na pulpicie operatorskim ten kierunek obrotów jest oznaczony jako CW, ang. *ClockWise*). Dla obrotów lewych jest odwrotny kierunek (oznaczenie na pulpicie CCW, ang. *CounterClockWise*). Konieczność włączenia lewych bądź prawych obrotów wynika z usytuowania narzędzia względem przedmiotu obrabianego i rodzaju tego narzędzia.

M6 – wymiana narzędzia.

Efektem działania tej funkcji jest pobranie narzędzia z magazynu narzędziowego i zamocowanie go w gnieździe narzędziowym, w którym znajduje się podczas obróbki tym narzędziem (patrz opis funkcji T). Jednocześnie narzędzie dotychczas tam się znajdujące zostaje przeniesione do magazynu narzędziowego (sterowanie

zmieniaczem narzędzi). Czasami procedura wymiany narzędzi jest zapisana w postaci podprogramu (opis w dokumentacji techniczno-ruchowej).

M8 – włączenie pompki chłodziwa;

M9 – wyłączenie pompki chłodziwa.

2.8. Inne elementy w programie sterującym

Dla zwiększenia czytelności programu sterującego często umieszcza się w nim komentarze, tj. pewne opisy słowne, które nie są analizowane przez układ sterowania. W języku Sinumerik 840D komentarzem jest zawartość bloku po znaku „;" aż do końca bloku, np.

```
N05 Z20 X30; to jest blok z komentarzem
```

W programach, w których stosuje się instrukcje strukturalne (pętle, rozgałęzienia) występują często etykiety bloków. Etykieta jest to ciąg znaków alfanumerycznych (zabronione jest używanie niektórych znaków – dokładne informacji w dokumentacji języka sterowania), zakończonych znakiem „:", znajdujących się na początku bloku. Dzięki temu jest możliwe wykonywanie skoków do bloków opatrzonych takimi etykietami (dokładniej zostanie to omówione w dalszej części instrukcji), np.

```
ETYKIETA1:  G0 X100 Y100 ; to jest blok z etykietą
....
N100 GOTOB ETYKIETA1    ; skok do bloku o podanej etykiecie
```

Projektując program sterujący można zamieścić w nim instrukcję powodującą wyświetlenie na pulpicie sterującym krótkiej informacji. Może ona np. informować operatora o konieczności wykonania czynności obsługowych po zatrzymaniu programu (funkcje M0/M1). Do tego celu służy funkcja **MSG** (ang. *MeSsaGe*) o następującej postaci argumentów (zaleca się stosować tę funkcję jako jedyną w bloku):

MSG(″Tekst informacji″)	– wyświetlenie tekstu informacji na pulpicie operatorskim (funkcja modalna);
MSG(″″)	– anulowanie wyświetlania tekstu na pulpicie.

2.9. Ogólna struktura bloku

Choć, jak wspomniano wcześniej, kolejność adresów w bloku nie ma większego znaczenia dla układu sterowania, na ogół przyjmuje się pewne uporządkowanie adresów w bloku (wynikające z przesłanek historycznych – taki układ adresów był obowiązkowy dla starszych układów sterowania). Nie jest ono obowiązkowe ale w zdecydowany sposób ułatwia analizę treści bloku. Wzorcowy blok może zatem mieć następującą postać:

N35 G90 G1 X100 Y100 F100 S500 T12 D1 M8 M4 **LF**

Gdzie kolejno umieszczane są w nim:

- ➔ numer bloku (N);
- ➔ funkcje przygotowawcze (G);
- ➔ współrzędne (adresy geometryczne X, Y, Z i inne);
- ➔ funkcje technologiczne (F, S);

➔ funkcje narzędziowe (T, D);
➔ funkcje pomocnicze (M).

Choć wcześniej wspomniano, iż blok jest w całości czytany z programu, analizowany i realizowany, to w rzeczywistości istnieje pewien priorytet wykonywania czynności, zaprogramowanych w bloku, np. wymiana narzędzia, ustawienie parametrów technologicznych, włączenie obrotów wrzeciona i wykonanie zaprogramowanego ruchu narzędziem, przy czym ruchy we wszystkich osiach sterowanych numerycznie (X, Y, Z) są wykonywane jednocześnie. **Kolejność ta nie wynika z kolejności adresów w bloku**. Przedstawiony powyżej blok może mieć również inną postać, np.:

N35 S500 G1 M8 X100 F100 T12 G90 D1 M4 Y100 **LF**

Ale taką postać bloku trudno zrozumieć i zanalizować.

Prezentowana kompletna struktura bloku na ogół jest rzadko używana, najczęściej w bloku występują tylko te adresy, które w danym bloku ulegają zmianie.

2.10. Ogólna struktura programu sterującego

Podobnie jak miało to miejsce dla bloku, również dla całego programu sterującego można wskazać preferowaną strukturę, choć oczywiście w praktyce można się spotkać z wieloma różnymi stylami treści programu sterującego. Taka uogólniona struktura programu może zatem wyglądać następująco:

```
%_N_0109867_MPF
;PROGRAM OBROBKI CZESCI 01-098-67
N5 G71 G90 G95 G54 G450
MSG(″TOCZENIE ZGRUBNE”)
N10 T1 D1 S1500 F200 M6
N15 G0 X100 Y100
N20 G1 X150
N25 Y120
.....................................
MSG(″KONIEC OBROBKI″)
N500 G53 T0 D0 G0 X500 Y600 Z450
MSG(″″)
N505 M30
```

Na początku programu powinna być umieszczona w postaci komentarza informacja o tym programie – opis przedmiotu obrabianego, nr rysunku, data utworzenia programu, nazwisko programisty itp. Początkowe bloki programu powinny zawierać wywołanie najważniejszych funkcji przygotowawczych (G), sterujących interpretacją programu (blok N5). Takie wywołanie, choć większość z tych funkcji już na starcie programu powinna być aktywowana przez układ sterowania, pozwala na lepsze zrozumienie programu. Może się także zdarzyć, iż na danej obrabiarce ustawienia domyślne funkcji przygotowawczych są inne niż standardowe.

Przed ciągiem bloków, programujących jakiś wyodrębniony fragment operacji (np. obróbka jednym narzędziem) zaleca się umieszczenie komentarza z opisem tego fragmentu (np. TOCZENIE ZGRUBNE). Rozpoczynając obróbkę nowym

narzędziem wyodrębnia się bloki przywołujące to narzędzie i parametry technologiczne (N10). Dopiero kolejne bloki zawierają instrukcje geometryczne, sterujące obróbką (N15, N20). Zaleca się podawać tylko te współrzędne, których wartości się zmieniają. **Stanowczo należy wystrzegać się programowania współrzędnych przed przywołaniem układu współrzędnych przedmiotu i korektorów narzędziowych** – grozi to kolizją przy wymianie narzędzi.

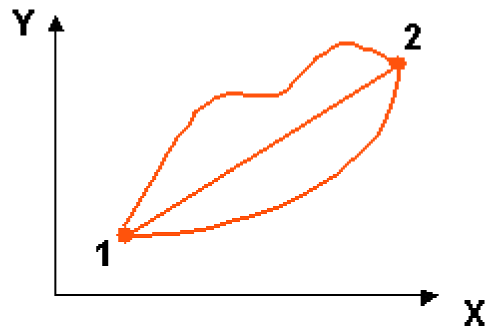
Na zakończenie programu powinno się zaprogramować zjazd zespołów ruchomych obrabiarki do pewnego stałego punktu, pozwalającego na bezpieczne wyjęcie przedmiotu obrabianego i założenie nowego, oraz na inne manipulacje w obrębie przestrzeni roboczej (N500). Położenie końcowe zespołów obrabiarki jest także położeniem początkowym w następnym wykonaniu tego samego lub innego programu, co ma duże znaczenie dla bezpiecznej pracy obrabiarki (uniknięcie kolizji). Podana w przykładzie sekwencja G53 T0 D0 (przywołanie punktu kodowego **M**, odwołanie korektorów narzędziowych) powoduje przejście do programowania we współrzędnych maszynowych, stąd podane współrzędne punktu odjazdu są niezależne od przyjętego w danym programie układu współrzędnych przedmiotu. Do odjazdu można wykorzystać także specjalnie do tego celu przeznaczone funkcje G74 lub G75 (informacje w następnych rozdziałach). Ostatni blok zawiera adres końca programu (M30 lub M2).

Podane w niniejszym rozdziale informacje, choć nie objaśniają wszystkich elementów programu sterującego, pozwalają już na wykonywanie pewnych prostych ćwiczeń (omówionych w następnym rozdziale), które w sposób stopniowy pozwalają na poznawanie wszystkich zawiłości procesu programowania obrabiarek CNC.

3. PROGRAMOWANIE RUCHÓW NARZĘDZI

3.1. Wiadomości ogólne

Zasadniczą częścią programu sterującego są bloki programujące ruch narzędzia. Aby w pełni opisać ten ruch wymagane są następujące dane (Rys. 45):

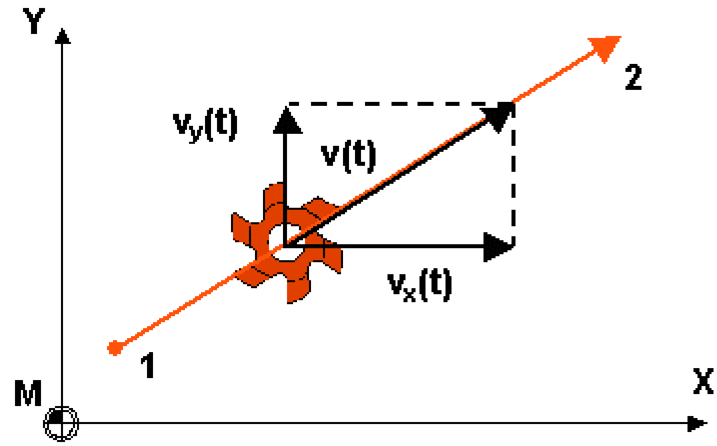


Rys. 45. Programowanie ruchu narzędzia

- ➔ Punkt początkowy ruchu (1);
- ➔ Punkt końcowy ruchu (2);
- ➔ Prędkość ruchu;
- ➔ Tor ruchu.

Idea sterowania numerycznego polega na programowaniu ruchu po torze ciągłym w ten sposób, że punkt końcowy ruchu w jednym bloku jest jednocześnie punktem początkowym ruchu w bloku następnym. Zatem w **bloku programuje się tylko punkt końcowy ruchu**. Prędkość ruchu jest programowana albo poprzez adres F (posuw) dla ruchu roboczego, albo pobierana z danych maszynowych (dla tzw. ruchu szybkiego).

Ostatnim elementem definicji ruchu jest tor ruchu, określany mianem interpolacji, tj. zachowaniem się punktu kodowego narzędzia pomiędzy programowanymi punktami. Należy ją rozumieć jako sposób powiązana programowego niezależnych ruchów w osiach maszynowych tak, aby uzyskać zamierzony wypadkowy tor przemieszczania się punktu kodowego narzędzia (Rys. 46). Nieco upraszczając to zagadnienie można powiedzieć, że na podstawie zadanego toru ruchu (czyli interpolacji) oraz zadanej prędkości ruchu układ sterowania jest w stanie obliczyć lokalną wartość prędkości $v(t)$, traktowaną jako wektor. Jest to oczywiście wartość wypadkowa, zatem układ sterowania za pomocą modułu (nazywanego dawniej interpolatorem) dokonuje wyznaczenia wektorów składowych prędkości w wymaganych osiach maszynowych (np. $v_x(t)$ i $v_y(t)$ na Rys. 46). To pozwala wygenerować z kolei sygnały sterujące napędami w poszczególnych osiach (jak wiadomo są one od siebie niezależne). Jeśli proces takich obliczeń będzie powtarzany z dostatecznie dużą częstotliwością (przy małych odstępach czasowych) to uzyska się tor ruchu w dużym przybliżeniu równy zadanemu. Podsumowując, interpolacja to zadanie toru ruchu narzędzia w postaci kodowego oznaczenia pewnej linii opisanej jej równaniem matematycznym (np. linia prosta, łuk koła, parabola, spirala, spline itp.)

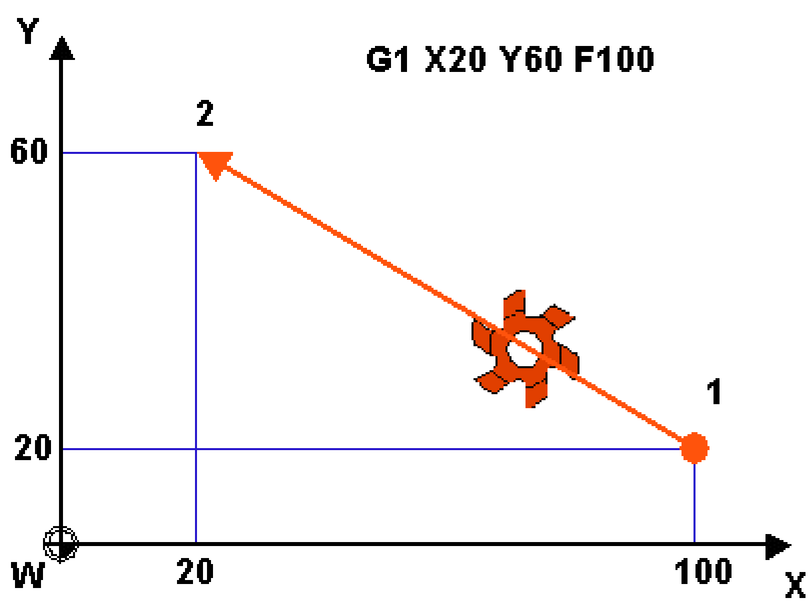


Rys. 46. Idea interpolacji

3.2. Interpolacja liniowa G1

Interpolacja liniowa – Rys. 47 – należy do najprostszych, a zarazem najczęściej wykorzystywanych ruchów roboczych (obróbczych). Tor ruchu narzędzia przebiega po linii prostej pomiędzy punktem początkowym i końcowym. Wymaga zaprogramowania posuwu (adres F) – podobnie jak pozostałe interpolacje robocze. Ruch roboczy związany jest również z większą dokładnością pozycjonowania w punkcie końcowym.

Interpolacja liniowa na tokarce pozwala toczyć powierzchnie czołowe, walcowe i stożkowe, na frezarce – wiercić, rozwiercać, frezować powierzchnie czołowe, wytaczać itp.

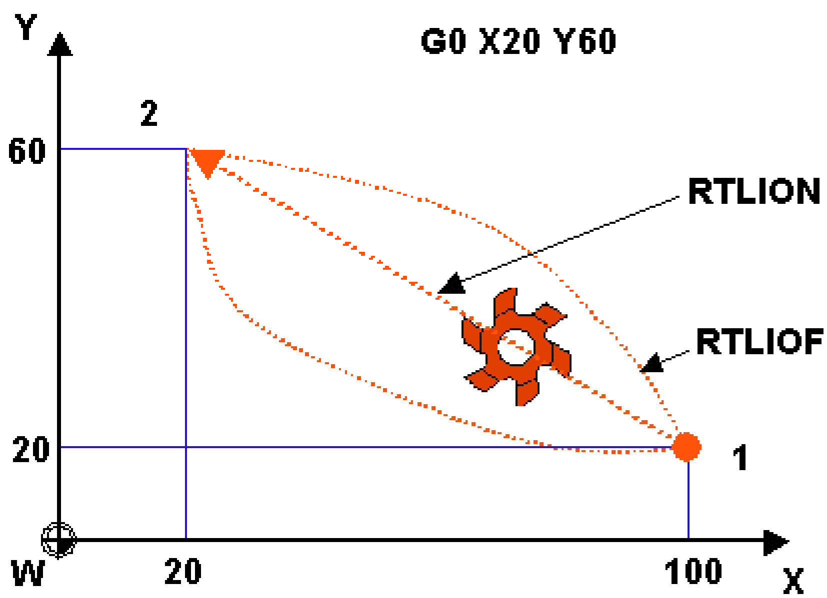


Rys. 47. Interpolacja liniowa G1

3.3. Interpolacja punktowa G0

Interpolacja punktowa (zwana też ruchem szybkim) – Rys. 48 – polega na przemieszczaniu się narzędzia do zaprogramowanego punktu końcowego z dużymi prędkościami w osiach sterowanych numerycznie. Ruch ten może być związany z brakiem powiązania ruchu w osiach, czyli de facto przy braku interpolacji (funkcja przygotowawcza **RTLIOF**, ang. *Rapid Tool Linear Interpolation Off*), czego efektem jest nieprzewidywalny tor ruchu narzędzia. Może też istnieć powiązanie ruchów w osiach, czyli interpolacja liniowa (funkcja przygotowawcza **RTLION**, ang. *Rapid Tool Linear Interpolation On*), czego efektem jest ruch narzędzia po linii prostej. Skutkiem ruchu szybkiego jest też zwiększona tolerancja dokładności pozycjonowania w punkcie docelowym (może wystąpić ruch oscylacyjny narzędzia wokół punktu końcowego ruchu z uwagi na bezwładność ruchomych zespołów obrabiarki).

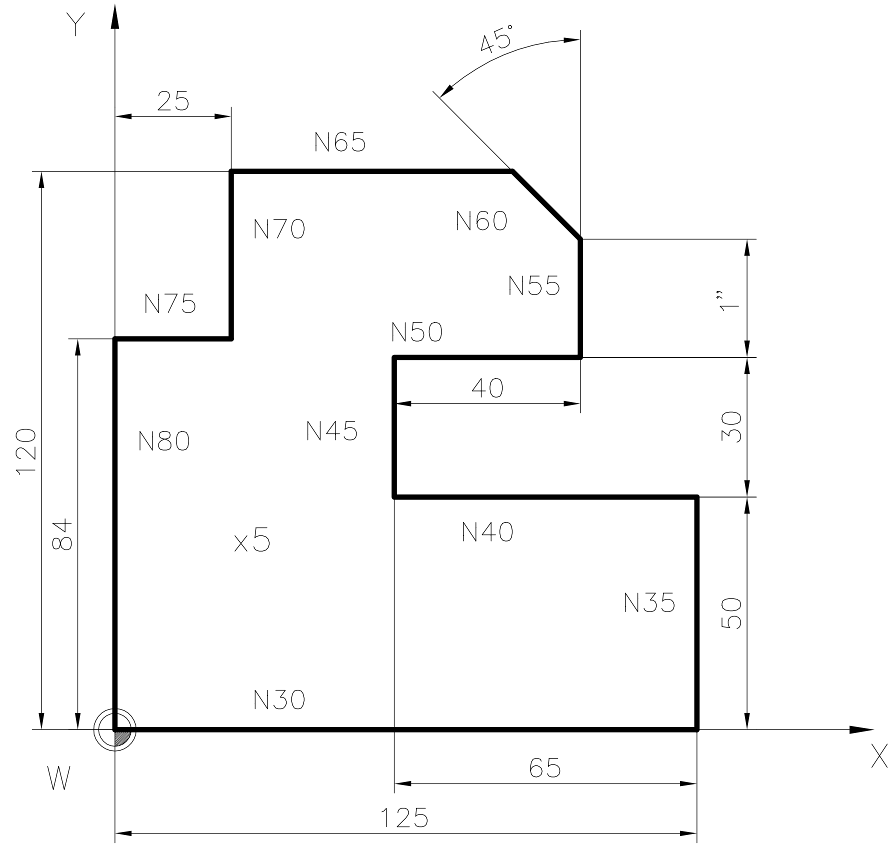
Interpolacja punktowa jest przeznaczona wyłącznie do ruchów ustawczych narzędzia. Należy pamiętać, że **skutkiem ruchu szybkiego może być kolizja** (kontakt narzędzia lub innego elementu ruchomego z przedmiotem obrabianym lub innym elementem obrabiarki). Z tego względu wszystkie ruchy z interpolacją punktową należy programować bardzo starannie.



Rys. 48. Interpolacja punktowa G0

3.3.1. Przykład

Zaprogramować ruch punktu kodowego narzędzia po konturze przedstawionym na Rys. 49, w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**. Ruch rozpocząć od punktu (0,0) w kierunku przeciwnym do ruchu wskazówek zegara. Przyjąć poziom materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 16 mm.



Rys. 49. Szkic przedmiotu do przykładu obróbki konturu z interpolacją liniową

Rozwiązanie:

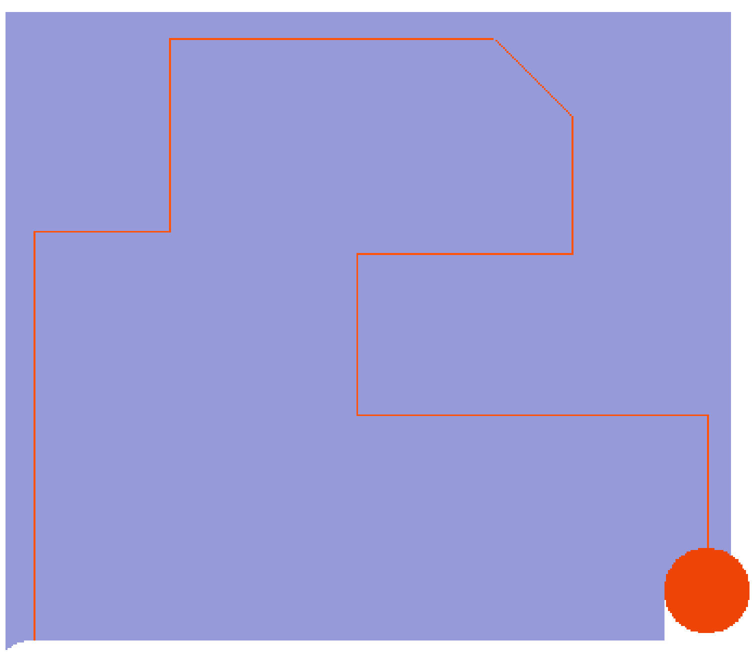
```
%_N_EX01_MPF
; 11-08-2003
N5 G40 G54 G71 G90 G94
N10 S800 F100 T1 D1 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 X125
N35 Y50
N40 X60
N45 Y80
N50 X100
N55 Y105.4
N60 X85.4 Y120
N65 X25
N70 Y84
N75 X0
N80 Y0
N85 G0 Z100
N90 G53 T0 D0 G0 X300 Y300 Z200 M9 M6 M5
N95 M30
```

W bloku N5 przywołano najważniejsze funkcje przygotowawcze, inicjując tym samym sposób interpretacji kolejnych bloków (funkcje te omówione zostaną w kolejnych rozdziałach). Blok N10 to przywołanie narzędzia (T1) w magazynie, wstawienie go do wrzeciona (M6) i ustawienie jego rejestru narzędziowego (D1).

W bloku tym dokonano ustawienia prędkości obrotowej wrzeciona na 800 obr/min (S800) i włączenie obrotów prawych (M3) oraz ustawienia prędkości posuwu na 100 mm/min (F100). Ponadto włączono pompkę chłodziwa (M8). Następne bloki programują już ruch narzędzia. W N15 następuje pozycjonowanie w płaszczyźnie XY nad punktem początkowym konturu (ruchem szybkim G0), a w N20 w osi Z 3 mm przed powierzchnią materiału. Blok N25 to ustawianie się w osi Z ruchem roboczym (G1) z uwagi na rozpoczęcie kontaktu narzędzia z materiałem obrabianym. Bloki N30÷N80 programują ruch narzędzia w płaszczyźnie XY, kolejno przez wszystkie elementy konturu (Rys. 49). Należy zauważyć, że nie wszystkie wymiary podane na rysunku pozwalają wprost przenieść je do programu sterującego. Część współrzędnych wymagała obliczeń co nie jest wygodne i może prowadzić do błędów. W dalszej części skryptu zostaną przedstawione metody takiego programowania współrzędnych aby wyeliminować obliczenia.

Blok N85 to wycofanie narzędzia w osi Z ruchem szybkim (G0). W przedostatnim bloku (N90) wyłączamy pompkę chłodziwa (M9) i wyłączamy obroty wrzeciona (M5), następnie wycofujemy narzędzie z wrzeciona do magazynu (M6), przy czym do wrzeciona nie wprowadzamy narzędzia nowego (zapewnia to specjalne słowo T0 – tzw. narzędzie zerowe). Odwołujemy programowanie w układzie współrzędnych przedmiotu – przejście na programowanie w układzie maszynowym (zapewnia to sekwencja słów G53 T0 D0). Ruchem szybkim (G0) pozycjonujemy zespoły obrabiarki w pewnym stałym położeniu wyjściowym (X300 Y300 Z200). Ostatni blok (N95) to słowo końca programu (M30).

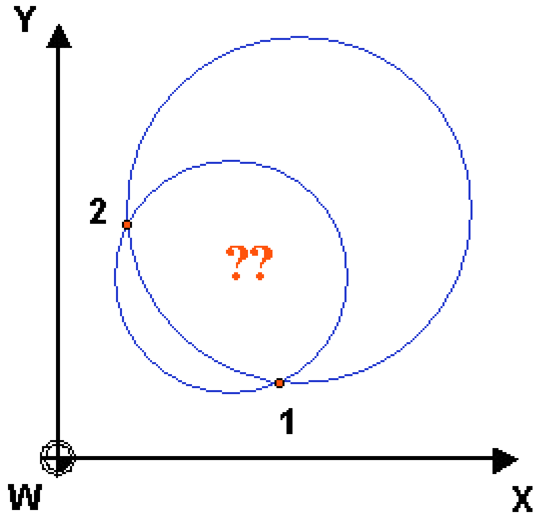
Po wykonaniu symulacji przedstawionego powyżej programu (Rys. 50) widać, że zaprogramowano ruch punktu środka freza, nie zaś obróbkę jego powierzchnią boczną. Program wymaga zatem wprowadzenia poprawek, które omówione zostaną w rozdziale poświęconym kompensacji promienia narzędzia.



Rys. 50. Symulacja programu sterującego dla przedmiotu z Rys. 49

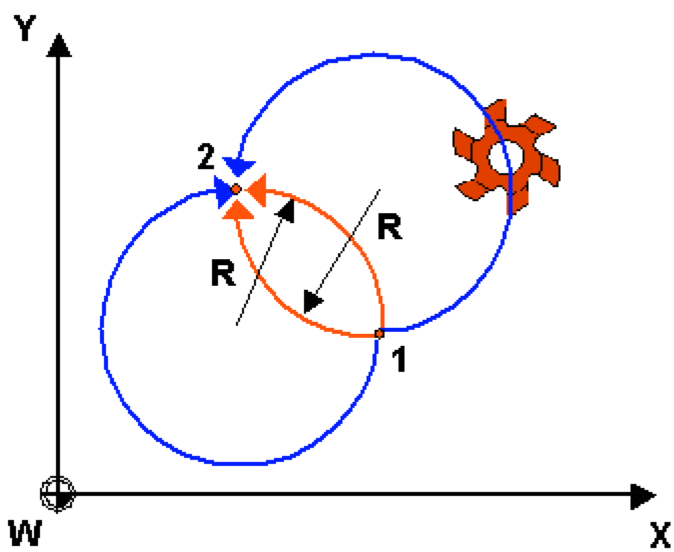
3.4. Interpolacja kołowa G2/G3

Ruch po łuku okręgu jest bardziej złożony niż miało to miejsce w przypadku interpolacji liniowej. Wynika to z faktu, iż okrąg nie może być jednoznacznie zdefiniowany przez podanie dwóch punktów (Rys. 51).



Rys. 51. Niejednoznaczność definicji ruchu z interpolacją kołową

Wymagane jest zatem podanie dodatkowych parametrów okręgu – najczęściej jest to promień. W takiej sytuacji można zbudować dwa okręgi o różnych położeniach środka, które dają cztery różne tory ruchu – po dwa po każdym okręgu (Rys. 52).

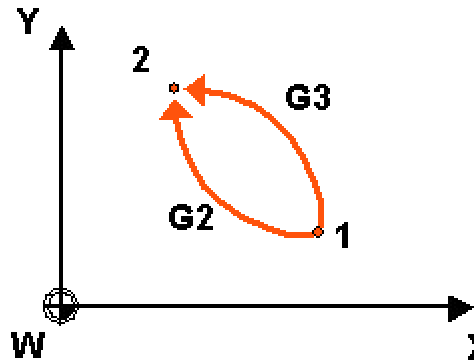


Rys. 52. Możliwe tory ruchu narzędzia z interpolacją kołową po okręgu o zadanym promieniu

Jeżeli przyjąć założenie, że z dwóch łuków dla jednego okręgu wybieramy ruch po łuku o mniejszej długości, to pozostaną nadal dwa różne tory (na Rys. 52 zaznaczone kolorem czerwonym) – stąd dla interpolacji kołowej przewidziano dwie funkcje:

G2 – interpolacja kołowa w kierunku zgodnym z ruchem wskazówek zegara;

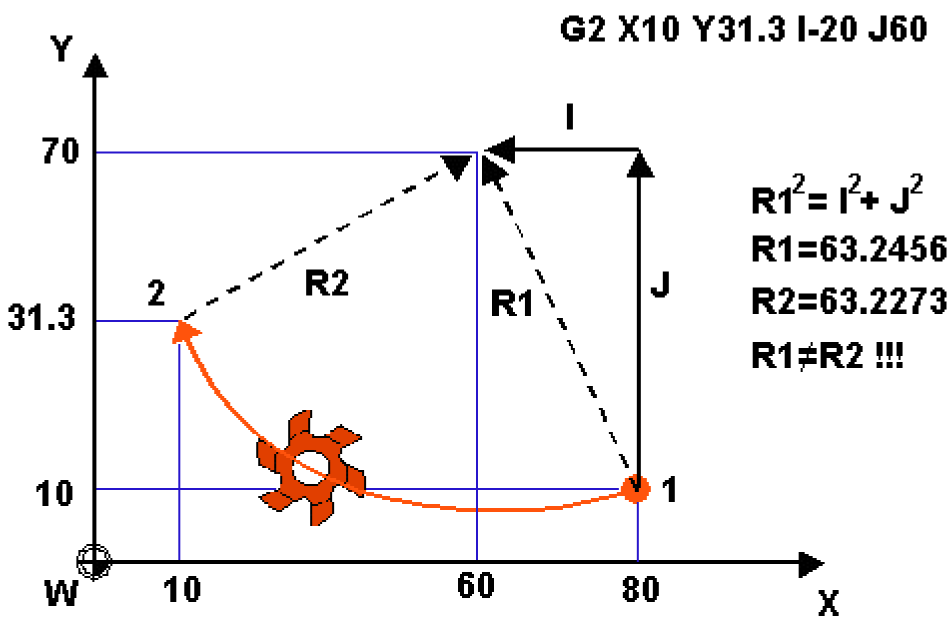
G3 **–** interpolacja kołowa w kierunku przeciwnym do ruchu wskazówek zegara (Rys. 53).



Rys. 53. Interpolacja kołowa G2/G3

W interpolacji kołowej przewidziano szereg różnych metod programowania promienia okręgu – w sposób bezpośredni lub pośredni. Najważniejsze z nich to:

1. Przyrostowe programowanie środka okręgu z wykorzystaniem niemodalnych parametrów interpolacji **I**, **J**, **K** – za ich pomocą programowany jest punkt środka okręgu; traktowane są one jako wektory składowe (w odpowiednich osiach – I w X, J w Y, K w Z) wektora od punktu początkowego ruchu do punktu środka okręgu – programowanie przyrostowe, niezależne od funkcji G90/G91. W tej metodzie promień okręgu jest wyznaczany przez układ sterowania z twierdzenia Pitagorasa (Rys. 54).

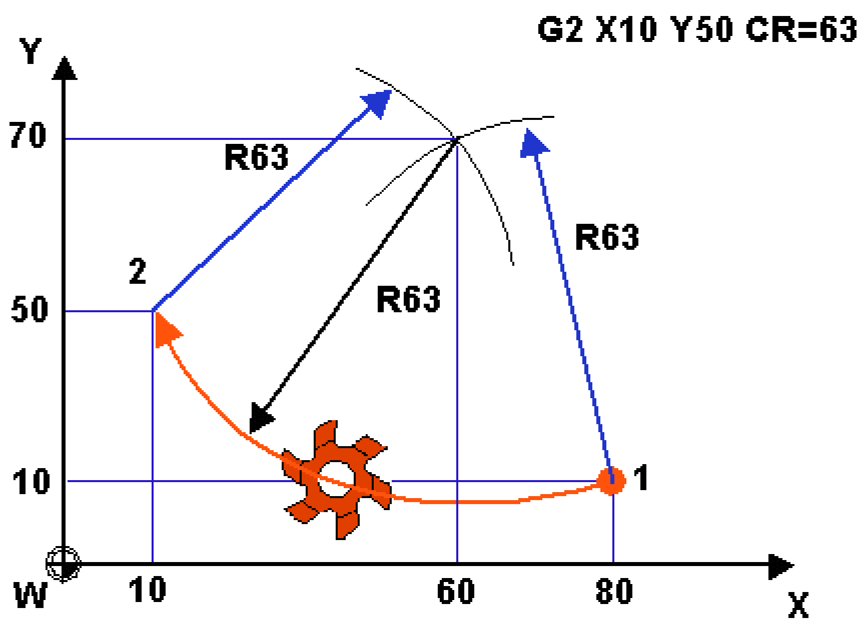


Rys. 54. Interpolacja kołowa G2/G3 z parametrami interpolacji I,J,K (przyrostowo)

Należy pamiętać, iż na wskutek przybliżonego wyznaczania promienia początkowego R1 (pierwiastkowanie) może się okazać, że jego długość jest różna od promienia końcowego R2 (Rys. 54). Układ sterowania zaakceptuje tą różnicę, o ile nie jest ona zbyt duża (dopuszczalna wartość różnicy jest zadana w danych

maszynowych układu sterowania), w przeciwnym przypadku wykonanie programu jest przerywane sygnalizacją błędu interpolacji kołowej.
Niniejsza metoda jest zalecaną, ponieważ w sposób bezpośredni zadany jest punkt środka okręgu, niezbędny dla realizacji interpolacji. W innych metodach programowania interpolacji kołowej układ sterowania musi sam obliczyć położenie środka okręgu.

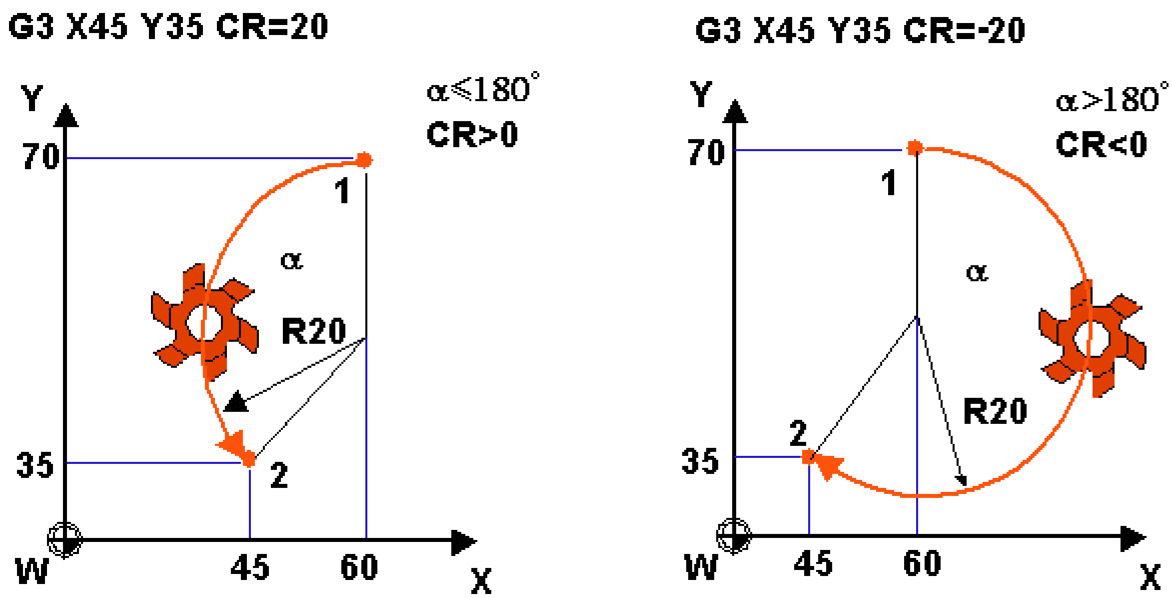
2. Bezpośrednie programowanie promienia okręgu (**CR,** ang. *Circle Radius*) – pod adresem CR podana jest wartość promienia okręgu. Układ sterowania na jego podstawie wylicza położenie punktu środka okręgu (Rys. 55).



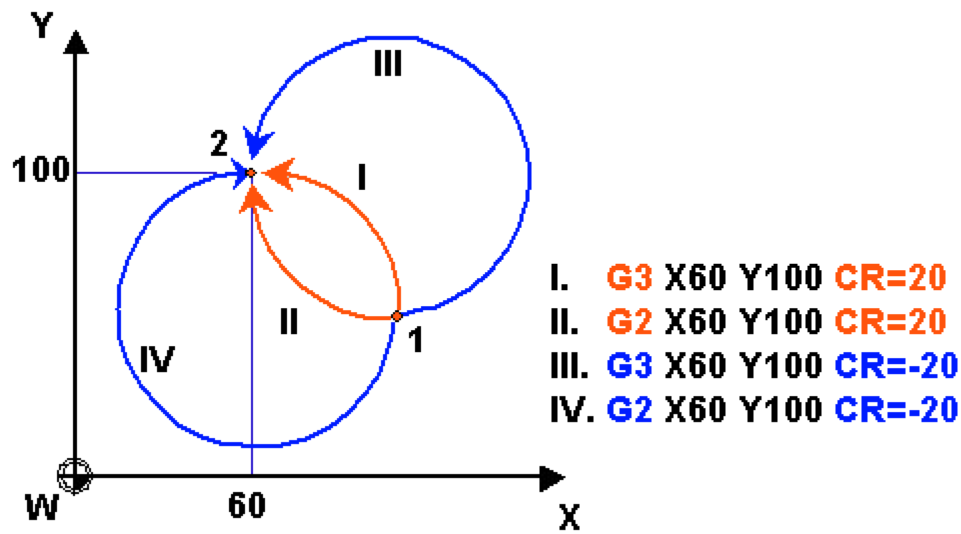
Rys. 55. Interpolacja kołowa z programowaniem promienia okręgu CR

Przy programowaniu promienia wartość adresu CR może być dodatnia lub ujemna. W zależności od tego układ sterowania wybiera tor ruchu narzędzia po krótszym bądź dłuższym łuku okręgu, co jest identyfikowane przez drogę kątową pomiędzy promieniem początkowym i końcowym. Dla wartości dodatniej adresu CR narzędzie wykonuje ruch po kącie równym lub mniejszym 180º, dla ujemnej – większym niż 180º – Rys. 56.

Efektem połączenia dwóch funkcji programowania interpolacji kołowej (G2, G3) z dwoma różnymi znakami adresu CR jest kombinacja czterech różnych torów ruchu narzędzia przy tej samej wartości promienia okręgu i tych samych punktach początkowym i końcowym łuku – Rys. 57.



Rys. 56. Zależność pomiędzy znakiem wartości adresu CR a torem ruchu narzędzia



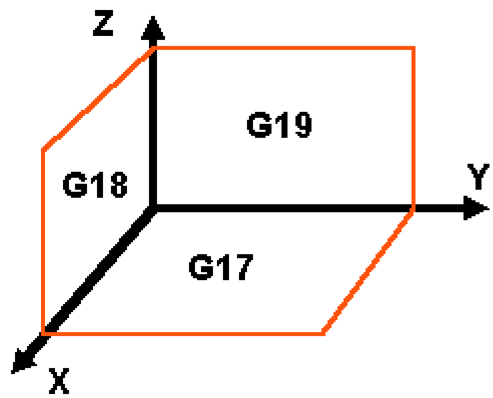
Rys. 57. Zależność toru ruchu od funkcji interpolacji G2/G3 i znaku wartości adresu CR

W przeciwieństwie do interpolacji liniowej czy punktowej, przy programowaniu interpolacji kołowej (również spiralnej czy w innych sytuacjach, opisanych w dalszych rozdziałach) należy pamiętać, że ma ona sens tylko w pewnej ustalonej płaszczyźnie. Przyjęto, że taką płaszczyznę musi definiować płaszczyzna aktualnego układu współrzędnych, przy czym wybór jednej z trzech możliwych odbywa się za pomocą funkcji przygotowawczych (stanowiących jedną grupę) – Rys. 58:

G17 – ustalenie płaszczyzny XY jako płaszczyzny interpolacji;
G18 – ustalenie płaszczyzny ZX jako płaszczyzny interpolacji;
G19 – ustalenie płaszczyzny YZ jako płaszczyzny interpolacji.

Dla tokarek domyślną funkcją jest G18, dla frezarek G17. Dla ułatwienia analizy pewnych zagadnień wprowadzono nazewnictwo osi uniezależnione od wybranej płaszczyzny interpolacji. Pierwsza z osi płaszczyzny interpolacji to oś odciętych (X dla

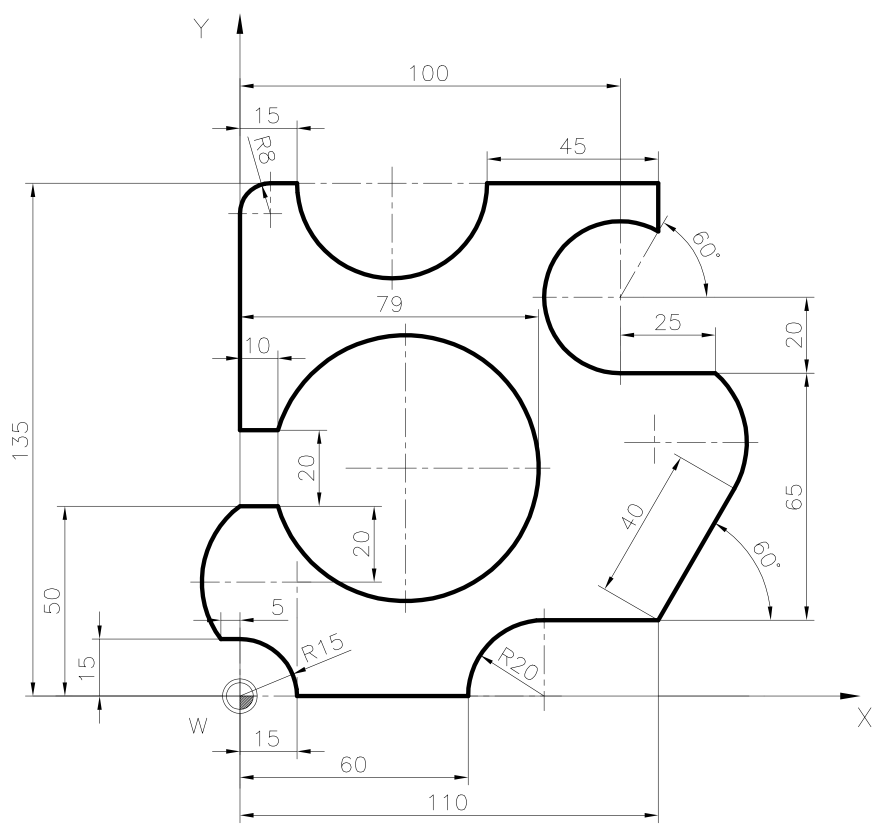
G17, Z dla G18, Y dla G19), druga to oś rzędnych, a oś prostopadła do płaszczyzny interpolacji – oś dosuwowa.



Rys. 58. Położenia płaszczyzn interpolacji i ich kodowanie

3.4.1. Przykład

Zaprogramować ruch narzędzia po konturze przedstawionym na Rys. 59 w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**. Ruch rozpocząć od punktu (0,0) w kierunku przeciwnym do ruchu wskazówek zegara. Przyjąć poziom materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 16 mm.



Rys. 59. Szkic przedmiotu do przykładu obróbki konturu z interpolacją kołową

Rozwiązanie:

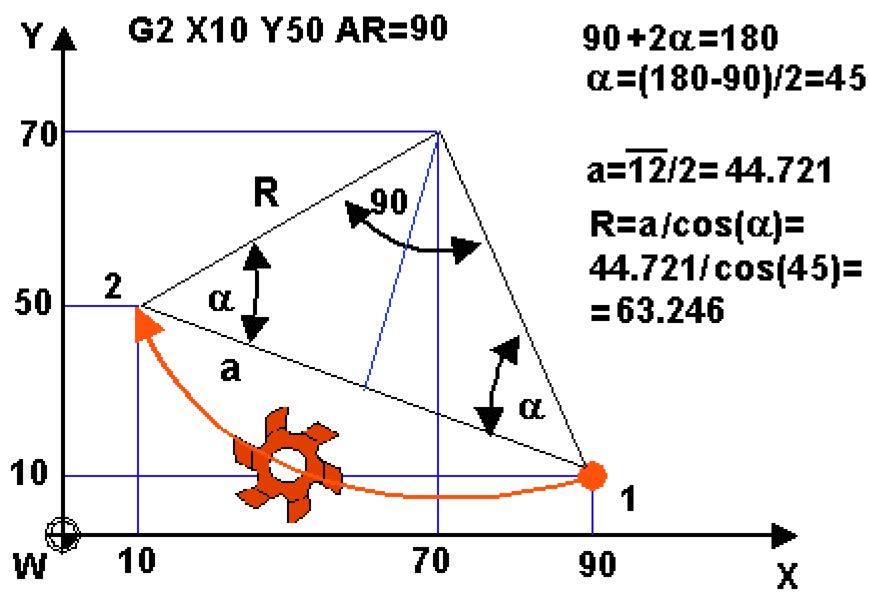
```
%_N_EX02_MPF
; 12-08-2003
N5 G40 G54 G71 G90 G94 G17
N10 S800 F100 T1 D1 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 X60
N35 G2 X80 Y20 CR=20
N40 G1 X110
N45 X130 Y54.641
N50 G3 X125 Y85 CR=24.262
N55 G1 X100
N60 G2 X110 Y122.321 CR=-20
N65 G1 Y135
N70 X65
N75 G2 X15 CR=25
N80 G1 X8
N85 G3 X0 Y127 CR=8
N90 G1 Y70
N95 X10
N100 G2 Y50 CR=-35
N105 G1 X0
N110 G3 X-5 Y15 CR=25
N115 G1 X0
N120 G2 X15 Y0 CR=15
N125 G0 Z100
N130 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N135 M30
```

Struktura programu jest podobna jak w poprzednim przykładzie (str. 51). Początkowe i końcowe bloki są identyczne (można wykorzystywać pewne szablony programów sterujących). Właściwa obróbka jest programowana w blokach N30÷N120. Podobnie jak w poprzednim przykładzie większość współrzędnych została obliczona (bądź odczytana z rysunku w systemie CAD) dlatego w następnych rozdziałach zostaną omówione inne metody zadawania współrzędnych, eliminujące tę niedogodność.

3.5. Inne metody programowania interpolacji kołowej

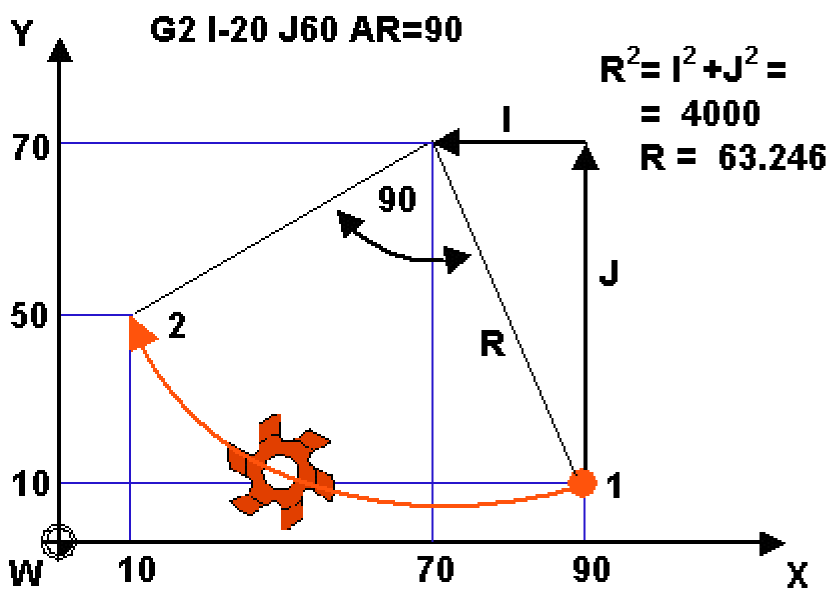
Podane w poprzednim rozdziale dwie metody programowania interpolacji kołowej nie wyczerpują wszystkich możliwych sposobów definicji ruchu po okręgu koła. Z pozostałych wymienić należy:

1. Programowanie kąta łuku za pomocą adresu **AR** – wyznaczenie parametrów łuku przy takim programowaniu przedstawiono na Rys. 60. Pod adresem AR jest programowana droga kątowa narzędzia po łuku, którego pozostałe parametry (środek, promień) są wyznaczane przez układ sterowania.



Rys. 60. Programowanie interpolacji kołowej przy użyciu kąta **AR**

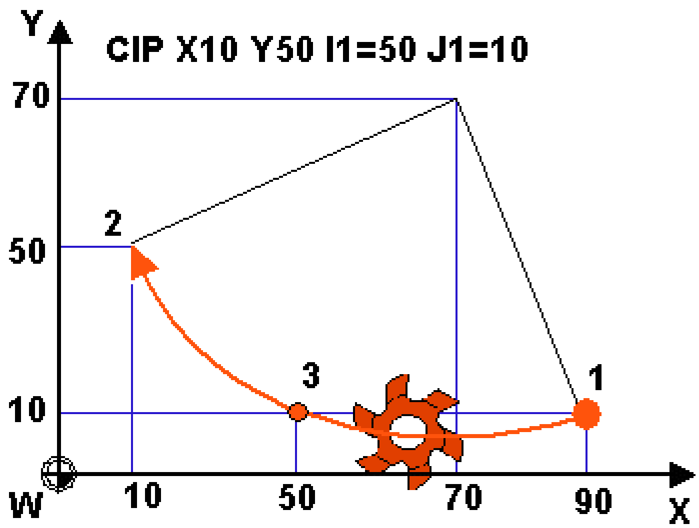
2. Programowanie środka i kąta łuku – w tym przypadku nie jest programowany punkt końcowy łuku (I,J,K), tylko położenie środka łuku i droga kątowa narzędzia (AR) – Rys. 61



Rys. 61. Programowanie interpolacji kołowej przy użyciu kąta **AR** i parametrów interpolacji **I,J,K**

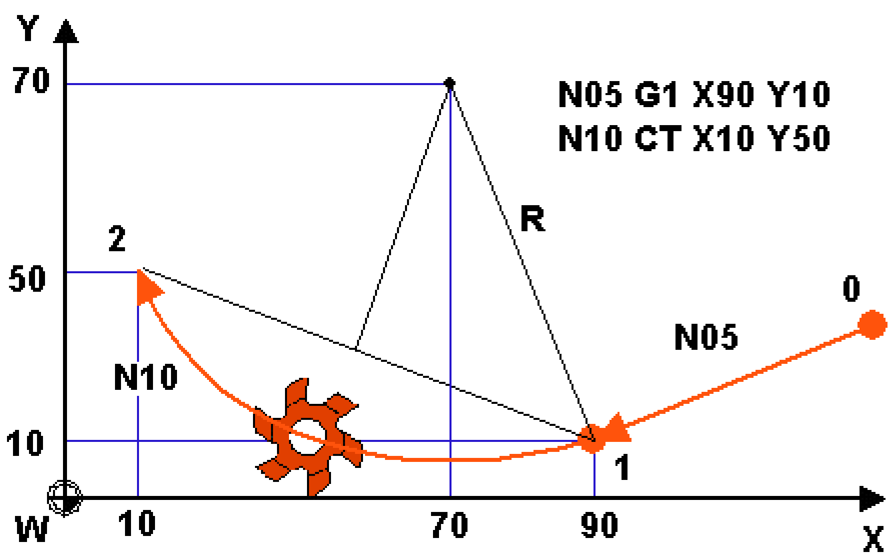
3. Łuk przez punkt pośredni **CIP** (ang. *Circle with Intermediate Point*) – w tej metodzie korzysta się z zasady, iż okrąg na płaszczyźnie jest jednoznacznie zdefiniowany przez trzy niewspółliniowe punkty. Programuje się zatem dodatkowy (pośredni) punkt ruchu po łuku okręgu, pomiędzy punktem początkowym i końcowym. Z uwagi na zasadę pojedyńczych wystąpień adresów w bloku współrzędne tego punktu programowane są pod adresami **I1, J1, K1** (odpowiednio

w osiach X, Y i Z). Zarówno kierunek interpolacji, jak i środek łuku wyznaczany jest przez układ sterowania (Rys. 62), stąd inne oznaczenie funkcji interpolacji.

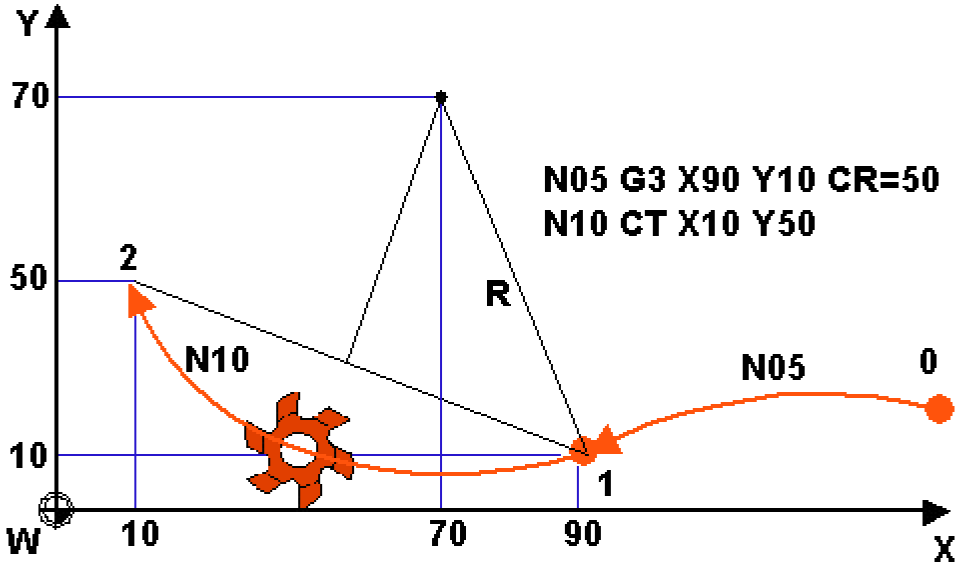


Rys. 62. Programowanie interpolacji kołowej przez punkt pośredni **CIP**

4. Łuk styczny **CT** (ang. *Circle Tangent*) – w tej metodzie programuje się jedynie punkt końcowy łuku, bez podania kierunku interpolacji i środka okręgu – jest to wyliczane przez układ sterowania na podstawie warunków styczności do poprzednio wykonywanego ruchu. Ruch ten może być ruchem z interpolacją liniową (Rys. 63) lub kołową (Rys. 64).

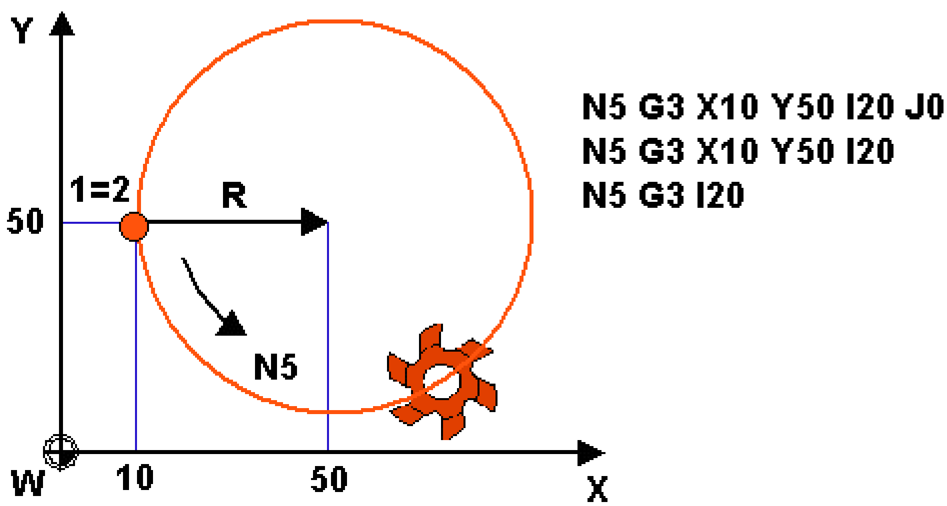


Rys. 63. Programowanie łuku stycznego (**CT**) do ruchu z interpolacją liniową (**G1**)



Rys. 64. Programowanie łuku stycznego (**CT**) do ruchu z interpolacją kołową (**G2/G3/CIP/CT**)

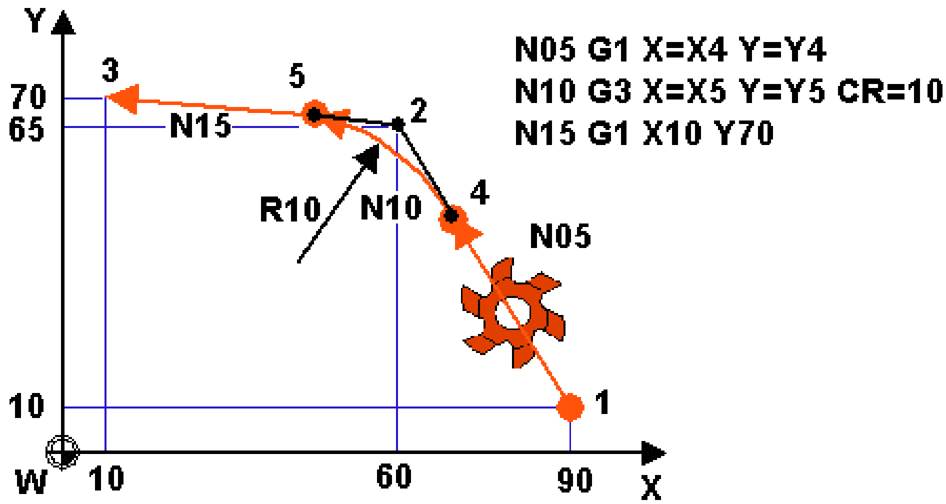
5. Programowanie ruchu po pełnym okręgu (**G2, G3**) – w tym wypadku układ sterowania zna współrzędne tylko jednego punktu, który jednocześnie jest punktem początkowym i końcowym ruchu. Dla jednoznacznego wyznaczenia parametrów ruchu niezbędne jest zaprogramowanie środka okręgu przez podanie parametrów **I, J, K**. Inne metody programowania (np. łuk styczny lub przez punkt pośredni) nie pozwalają na jednoznaczne wyznaczenie położenia środka okręgu. Na Rys. 65 podano kilka sposobów programowania tego ruchu, korzystając z różnych sposobów zadawania współrzędnych.



Rys. 65. Programowanie ruchu po pełnym okręgu

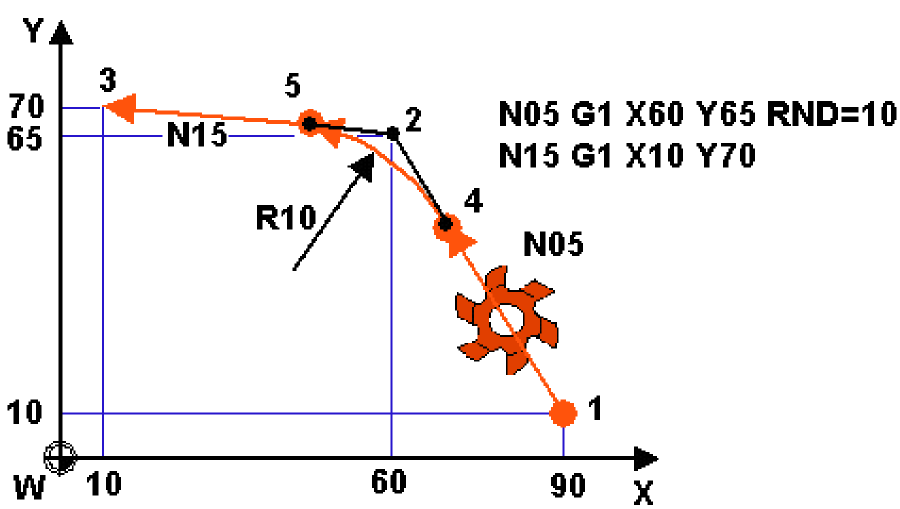
6. Programowanie łuków stycznych przy przejściu między odcinkami linii prostych (**RND, RNDM**) – w wielu przedmiotach obrabianych wykonuje się stępienie krawędzi poprzez wykonanie zaokrąglenia stycznego lub sfazowania, występującego najczęściej pomiędzy odcinkami linii prostych. Gdyby obróbkę

zaokrąglenia stycznego programować przy przyjęciu ogólnie obowiązujących zasad, to wymagałaby ona trzech bloków (Rys. 66).



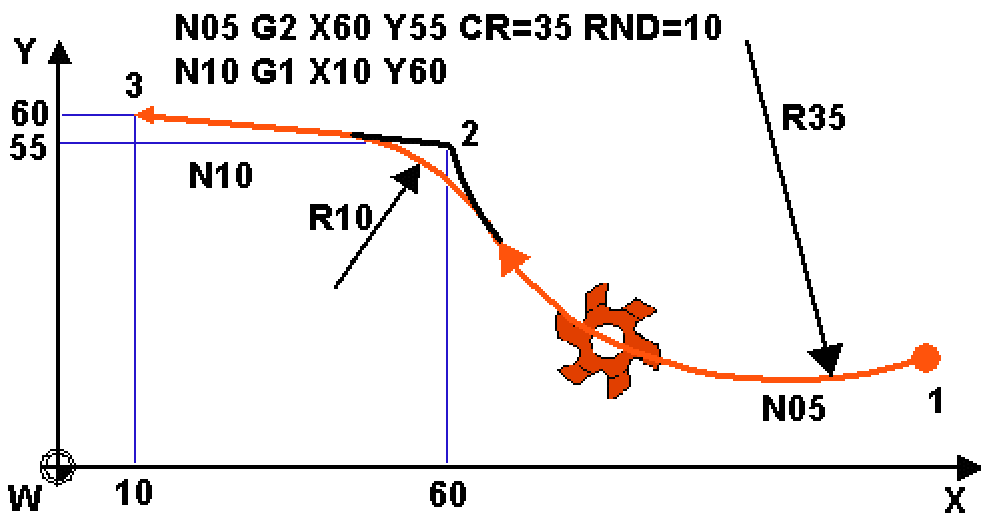
Rys. 66. Programowanie zaokrąglenia krawędzi bez wykorzystania specjalnych funkcji

Nie zawsze podane są współrzędne punktów styczności (4 i 5 na Rys. 66), które należałoby obliczyć. Stąd też wprowadzono możliwość programowania tego konturu przy użyciu tylko dwóch bloków, programujących ruch do tzw. punktu pozornego przecięcia (2), najczęściej zwymiarowanego na rysunkach konstrukcyjnych. Pomija się zatem drugi blok, programujący interpolację kołową (**G2/G3**), zastępując go adresem **RND** (ang. *RouNDed*) lub **RNDM** (ang. *RouNDed Modal*), które określaja promień zaokrąglenia pomiędzy programowanymi odcinkami konturu, umieszczając go w bloku opisującym pierwszy z nich (Rys. 67). Różnica między obu funkcjami polega na tym, iż RND jest niemodalna (wymagana jest w każdym bloku po którym wykonywane jest zaokrąglenie), zaś RNDM modalna – po przywołaniu niezerowej wartości adresu zaokrąglenie zostanie wykonane w każdym bloku z interpolacją liniową bądź kołową aż do odwołania funkcji (z zerową wartością lub brakiem wartości).



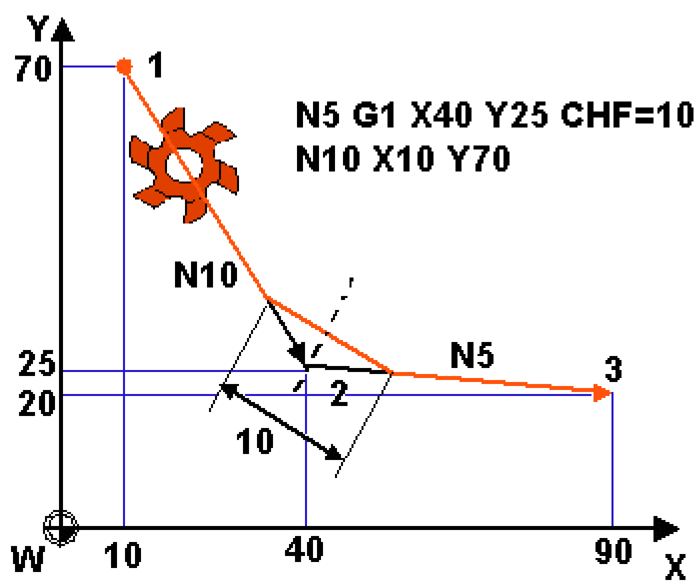
Rys. 67. Programowanie zaokrąglenia krawędzi z wykorzystaniem funkcji **RND**

W trakcie analizy programu układ sterowania obliczy współrzędne punktów styczności, natomiast podczas wykonywania pierwszego z bloków narzędzie wykona ruch do pierwszego punktu styczności (4), w drugim zaś bloku – promień zaokrąglenia do drugiego punktu styczności (5) oraz ruch do punktu końcowego (3). Styczne zaokrąglenie konturu może być wykonane pomiędzy odcinkami z interpolacją liniową i/lub kołową (Rys. 68).

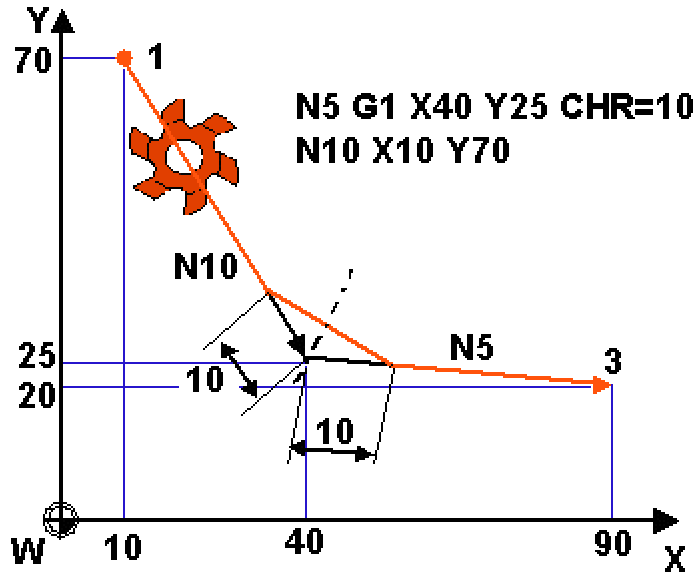


Rys. 68. Programowanie zaokrąglenia krawędzi z wykorzystaniem funkcji RND pomiędzy łukiem a odcinkiem linii prostej

Na zasadzie podobnej jak styczne promienie zaokrągleń są programowane sfazowania krawędzi, przy czym musi być spełniony warunek symetryczności fazy, tj. równej szerokości sfazowania krawędzi. Korzysta się z dwóch dostępnych adresów – **CHF** i **CHR** (ang. *Chamfer*). Pierwszy z nich definiuje długość fazy (Rys. 69), drugi – jej szerokość (Rys. 70).



Rys. 69. Programowanie sfazowania krawędzi z wykorzystaniem funkcji **CHF** (długość fazy)



Rys. 70. Programowanie sfazowania krawędzi z wykorzystaniem funkcji **CHR** (szerokość fazy)

3.5.1. Przykład

Rozwiązać przykład obróbki konturu (rozdz. 3.4.1) korzystając z opisanych funkcji programowania interpolacji kołowej.

Rozwiązanie:

```
%_N_EX02_MPF
; 12-08-2003
N5 G40 G54 G71 G90 G94 G17
N10 S800 F100 T1 D1 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 X60
N35 G2 I20 J0 AR=90
N40 G1 X110
N45 X130 Y54.641
N50 CT X125 Y85
N55 G1 X100
N60 G2 J20 AR=210
N65 G1 Y135
N70 X65
N75 G2 X15 AR=180
N80 G1 X0 RND=8
N90 G1 Y70
N95 X10
N100 CIP Y50 I1=79 J1=60
N105 G1 X0
N110 G3 X-5 Y15 CR=25
N115 G1 X0
N120 G2 X15 Y0 CR=15
N125 G0 Z100
N130 G53 T0 D0 G0 X300 Y300 Z200 M9 M5 M30
```

4. UKŁADY WSPÓŁRZĘDNYCH – DEFINICJE, TRANSFORMACJE

W praktyce programowania rzadko można spotkać sytuację, kiedy cały program sterujący opiera się na jednym, absolutnym, kartezjańskim układzie współrzędnych przedmiotu (WKS). Umiejętność dobrego programowania to przede wszystkim sprawność w operowaniu różnego rodzaju typami i przekształceniami układów współrzędnych. Wynika to faktu, iż wymiarowanie na rysunku konstrukcyjnym, będącym najczęściej podstawą do wyznaczania wartości współrzędnych do programu, nie zawsze pozwala wprost te współrzędne pobrać z rysunku. Z kolei wykonywanie obliczeń, nawet z użyciem kalkulatora czy komputera, jest kłopotliwe i może przyczynić się do powstania błędów obróbki, wynikających choćby z tolerowania wymiarów. Dużym problemem jest również wprowadzanie zmian w wymiarowaniu przedmiotu obrabianego. W takiej sytuacji trzeba raczej trudno szukać zmienonych wartości w programie i właściwe jedynym rozwiązaniem jest wtedy pisanie go od początku. Rozwiązaniem idealnym byłby zatem taki zapis programu sterującego, w którym zawarte były wszystkie wymiary przeniesione z rysunku konstrukcyjnego, co znacznie ułatwia jego analizę, weryfikację i modyfikację. Niniejszy rozdział opisuje jak taki stan osiągnąć.

4.1. Programowanie w układzie współrzędnych przedmiotu

Przed rozpoczęciem programowania w układzie współrzędnych przedmiotu należy ten fakt zasygnalizować układowi sterowania. Konkretnie należy wybrać rejestr przesunięć punktu zerowego (PPZ), transformującego układ maszynowy (MKS) w układ przedmiotu, nazywane też nastawnymi przesunięciami punktu zerowego. Standardowo Sinumerik zawierają cztery takie rejestry, tym niemniej można dodać kolejne. Wybór aktualnego układu współrzędnych jest realizowany za pomocą następujących modalnych funkcji przygotowawczych:

G500 – wyłączenie wszystkich przesunięć punktu zerowego – programowanie względem punktu maszynowego **M**;
G54 – przywołanie 1. rejestru przesunięć punktu zerowego;
G55 – przywołanie 2. rejestru przesunięć punktu zerowego;
G56 – przywołanie 3. rejestru przesunięć punktu zerowego;
G57 – przywołanie 4. rejestru przesunięć punktu zerowego.

Możliwe jest użycie dodatkowych rejestrów PPZ (5., 6.,...,99.), programowanych pod adresami **G505**, **G506**,..., **G599.** Użycie powyższych adresów jest równoważne z kasowaniem wszelkich transformacji układów współrzędnych (FRAMES) – rozdz. 4.5. Dodatkowo, przewidziano niemodalną (lokalną, aktywną w jednym bloku) funkcję **G53** wyłączania rejestrów PPZ (odpowiednik funkcji **G500**).

Przy programowaniu w układzie WKS należy jasno sprecyzować jak ten układ jest zorientowany względem przedmiotu obrabianego (np. za pomocą szkicu), co pozwoli dokładnie go zdefiniować na obrabiarce.

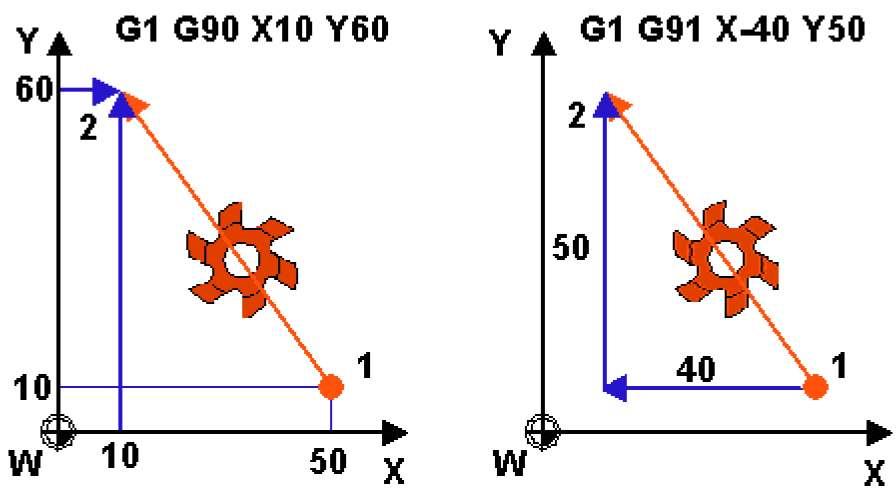
4.2. Definiowanie rodzaju i jednostek współrzędnych

4.2.1. Współrzędne absolutne i przyrostowe

Domyślnym układem współrzędnych przedmiotu jest układ absolutny prostokątny, tzn. taki, gdzie współrzędne są odnoszone do jednego, ustalonego punktu zerowego (**W**). Tymczasem na rysunkach konstrukcyjnych wymiarowanie rzadko prowadzi się względem jednej, wspólnej bazy wymiarowej. Często wymiary mają charakter przyrostowy, w postaci łańcucha wymiarowego. Wtedy wartości współrzędnych w układzie absolutnym byłyby obliczane, co przy wymiarach tolerowanych łatwo prowadzi do błędów. Stąd w układach sterowania przewidziano również możliwość programowania przyrostowego (inkrementalnego) względem aktualnego położenia narzędzia. Do obsługi trybu absolutnego i przyrostowego wymiarowania przewidziano cztery funkcje:

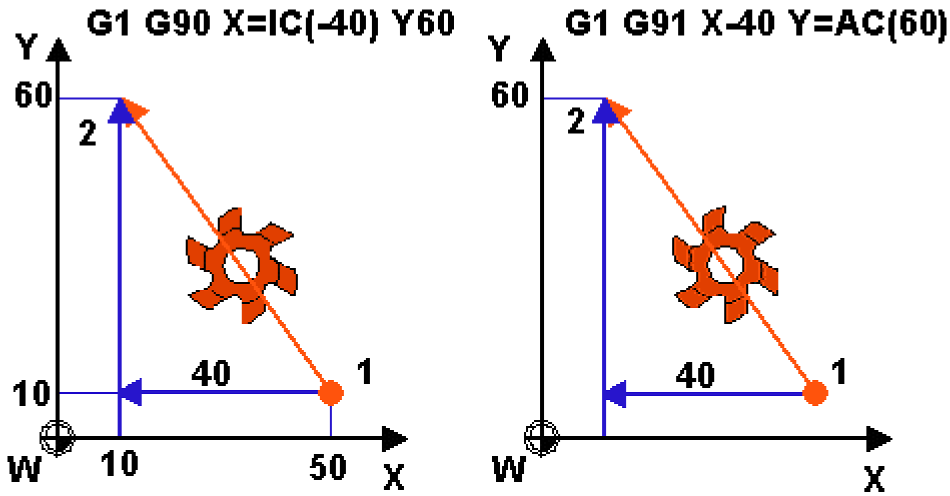
G90 – programowanie absolutne (funkcja modalna);
G91 – programowanie przyrostowe (funkcja modalna);
AC – programowanie absolutne (ang. *Absolute Coordinate*, funkcja niemodalna);
IC – programowanie przyrostowe (ang. *Incremental Coordinate*, funkcja niemodalna);

W programowaniu absolutnym (Rys. 71) wartość wymiaru odnosi się do aktualnego położenia punktu zerowego układu współrzędnych. W programowaniu przyrostowym wartość wymiaru odnosi się do aktualnego położenia narzędzia – jest ono traktowane jako chwilowe położenie punktu zerowego układu współrzędnych.



Rys. 71. Współrzędne w układzie absolutnym i przyrostowym

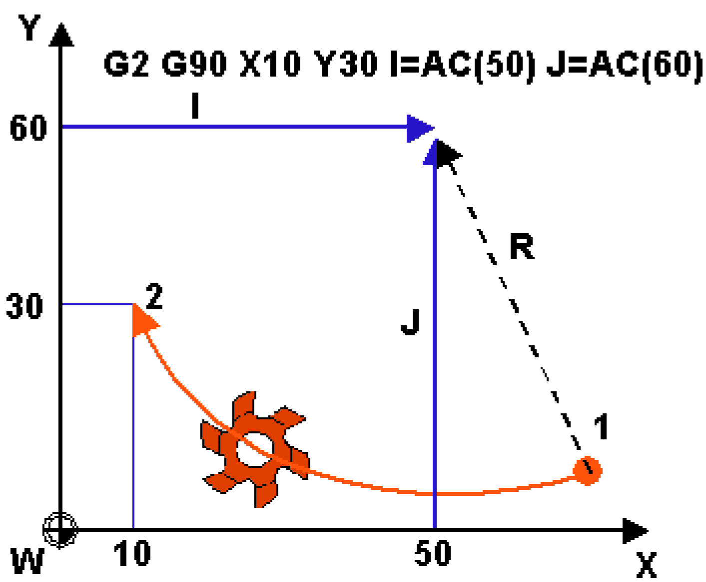
Funkcje modalne (globalne) odnoszą się do całego bloku. Jednak zdarza się czasem, że byłoby wygodnie część współrzędnych w bloku podać inny sposób niż reszta bloku. W tej sytuacji stosuje się funkcje niemodalne, odnoszące się do pojedynczych adresów (Rys. 72). Mogą one zostać użyte zarówno w stosunku do adresów wyrażających współrzędne liniowe, jak i kątowe (np. w programowaniu biegunowym – rozdz. 1.3).



Rys. 72. Mieszany sposób podawania współrzędnych w układzie absolutnym i przyrostowym

Sposób traktowania innych wymiarów, zadawanych w programie sterującym, może być bardzo różny. Niektóre z nich niezależnie od funkcji G90/G91 są podawane przyrostowo (np. parametry interpolacji I, J, K), inne zaś absolutnie (np. adresy CR, RND, CHF, CHR). Do zmiany tego stanu wykorzystuje się funkcje niemodalne (AC, IC). Jako przykład można podać jeszcze dwa kolejne sposoby programowania interpolacji kołowej:

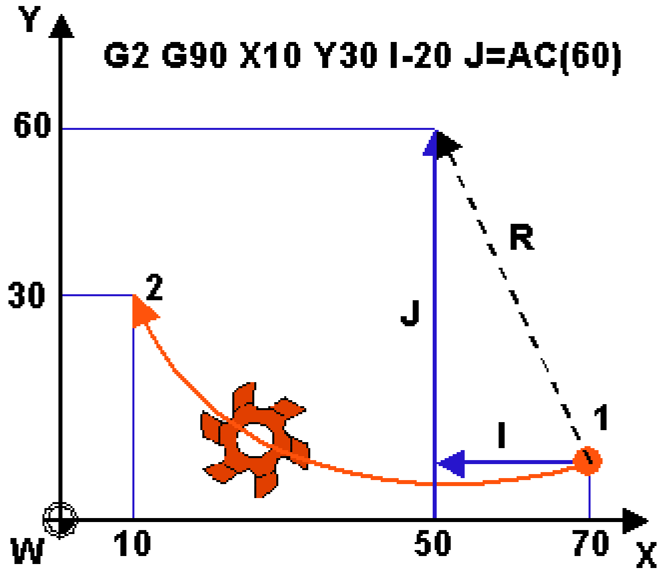
1. Absolutne programowanie położenia środka okręgu z wykorzystaniem niemodalnych parametrów interpolacji **I**, **J**, **K** – za ich pomocą programowany jest punkt środka okręgu (absolutnie); traktowane są one jako wektory składowe (w odpowiednich osiach) wektora od punktu zerowego aktualnego układu współrzędnych do punktu środka okręgu – Rys. 73.



Rys. 73. Interpolacja kołowa z absolutnym wymiarowaniem położenia środka łuku

2. Mieszane programowanie położenia środka okręgu z wykorzystaniem niemodalnych parametrów interpolacji **I**, **J**, **K** – za ich pomocą programowany jest

punkt środka okręgu, przy czym oba parametry interpolacji mogą być programowane w różny sposób z wykorzystaniem funkcji niemodalnych AC – Rys. 74.



Rys. 74. Interpolacja kołowa z mieszanym wymiarowaniem położenia środka łuku

4.2.2. Jednostki

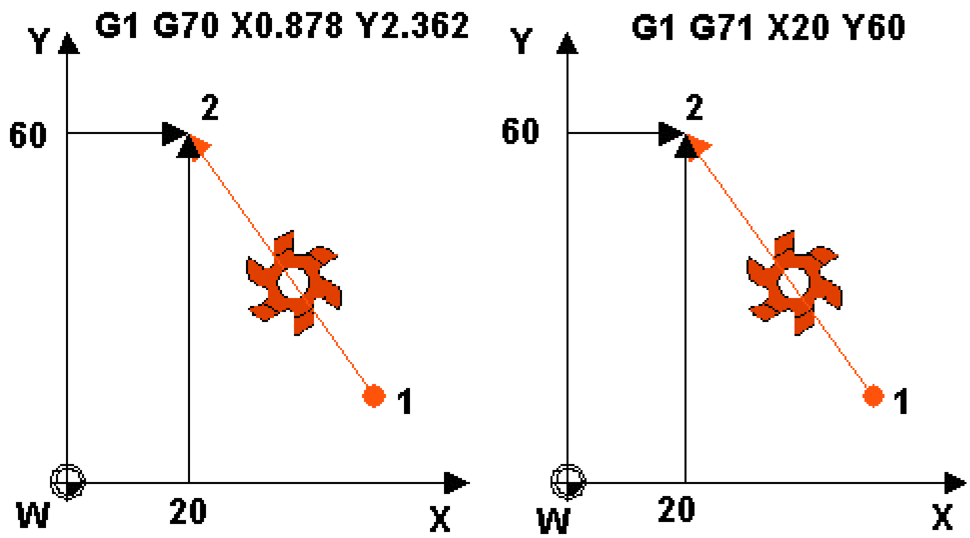
Wielkości kątowe (np. współrzędne w osiach A, B, C) przyjęto podawać w stopniach (kąt pełny to 360°). Inaczej jest ze współrzędnymi liniowymi (np. w osiach X, Y, Z), które praktycznie mogą być wyrażone w dowolnych jednostkach. Za jednostkę podstawową przyjęto [mm], jest to jednocześnie wewnętrzna jednostka układu sterowania, w której wykonuje obliczenia, odczytuje aktualne położenia z układów pomiarowych oraz przekazuje sygnały sterujące do układów napędowych. Programista chcąc wykorzystywać jednostkę podstawową lub inną jednostkę wymiarów liniowych (np. [cm]) musi poinformować o tym układ sterowania za pomocą funkcji przygotowawczych:

G70 **–** programowanie w jednostkach dodatkowych;

G71 – programowanie w jednostkach podstawowych (Rys. 75).

Jeżeli aktywna jest pierwsza z funkcji (G70) układ sterowania pobiera z danych maszynowych mnożnik, pozwalający mu przeliczenie jednostek dodatkowych na podstawowe. Standardowo wartość tego mnożnika wynosi 25,4 (stąd często funkcja G70 nazywana jest programowaniem w calach). Funkcje G70 i G71 odnoszą się do wymiarów geometrycznych, programowanych pod adresami X, Y, Z, I, J, K oraz dodatkowych adresów, np. promieni zaokrągleń itp.

Podane funkcje nie wpływają na jednostki posuwu (może być wyrażony zarówno w [mm/min] jak i [calach/min]), które są ustalone przez dane maszynowe z pulpitu operatorskiego. Nie wpływają również na wartości długości korekcyjnych narzędzi (wyrażonych zawsze w [mm]).



Rys. 75. Współrzędne wyrażone w jednostkach podstawowych i dodatkowych

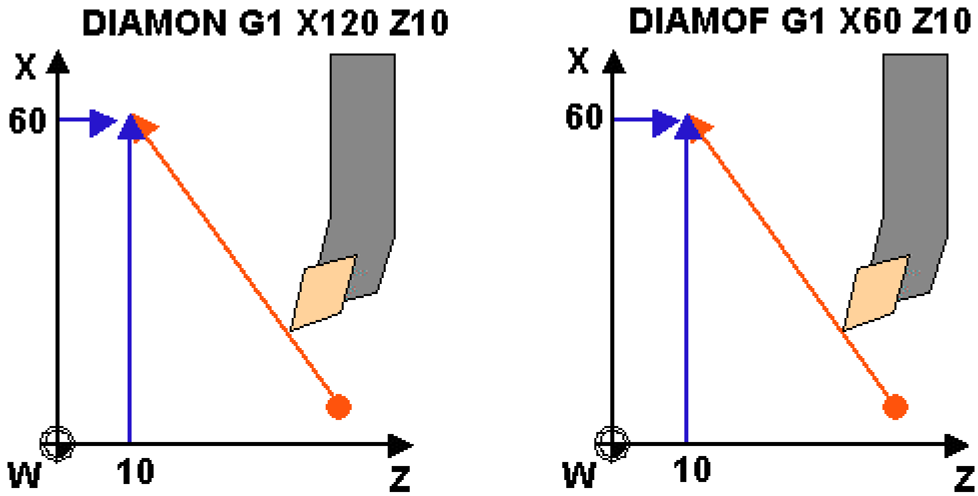
4.2.3. Wymiary średnicowe i promieniowe

Kolejnym zagadnieniem dotyczącym wymiaru programowanych współrzędnych jest wymiarowanie średnicowe i promieniowe. Pojawia się ono przede wszystkim w obróbce tokarskiej. Większość wymiarów w osi X na rysunkach konstrukcyjnych jest podawana średnicowo. Gdyby dosłownie potraktować współrzędne w osi X (Rys. 31) jako promieniowe, to pisząc program wymiary z rysunku należałoby dzielić przez dwa. Jednak jak to przedstawiono na Rys. 32 można traktować wymiary w osi X jako średnicowe, przy czym wewnętrznie układ sterowania dzieli je przez dwa i operuje na współrzędnych promieniowych. Należy jeszcze wspomnieć o tym, iż operację dzielenia współrzędnych można wykonywać dla dowolnej osi, przy czym w układzie sterowania (a konkretnie w danych maszynowych) musi ona być odpowiednio opisana (szczegóły w dokumentacji układu sterowania).

Do informowania układu sterowaniu o sposobie traktowania współrzędnych w osiach o wymiarowaniu średnicowym służą następujące funkcje:

DIAMON – (ang. *DIAMeter ON*) wymiary podawane średnicowo;
DIAMOF – (ang. *DIAMeter OFf*) wymiary podawane promieniowo;
DIAM90 – wymiary średnicowo przy programowaniu absolutnym (G90), promieniowo przy przyrostowym (G91).

Wymiarowanie średnicowe (Rys. 76) jest charakterystyczne dla tokarek i tam funkcja DIAMON jest domyślnie aktywna, wymiarowanie promieniowe (DIAMOF) z kolei aktywne jest dla frezarek. Należy pamiętać, że funkcja DIAMON odnosi się tylko i wyłącznie do współrzędnych, a nie do pozostałych wartości, np. parametrów interpolacji I, J, K (zawsze są podawane promieniowo).



Rys. 76. Wymiarowanie średnicowe i promieniowe

4.2.4. Przykład dla obróbki frezarskiej

Dla obróbki frezowania konturu z interpolacją kołową (rozdz. 3.4.1) wykorzystać poznane funkcje zadawania współrzędnych.

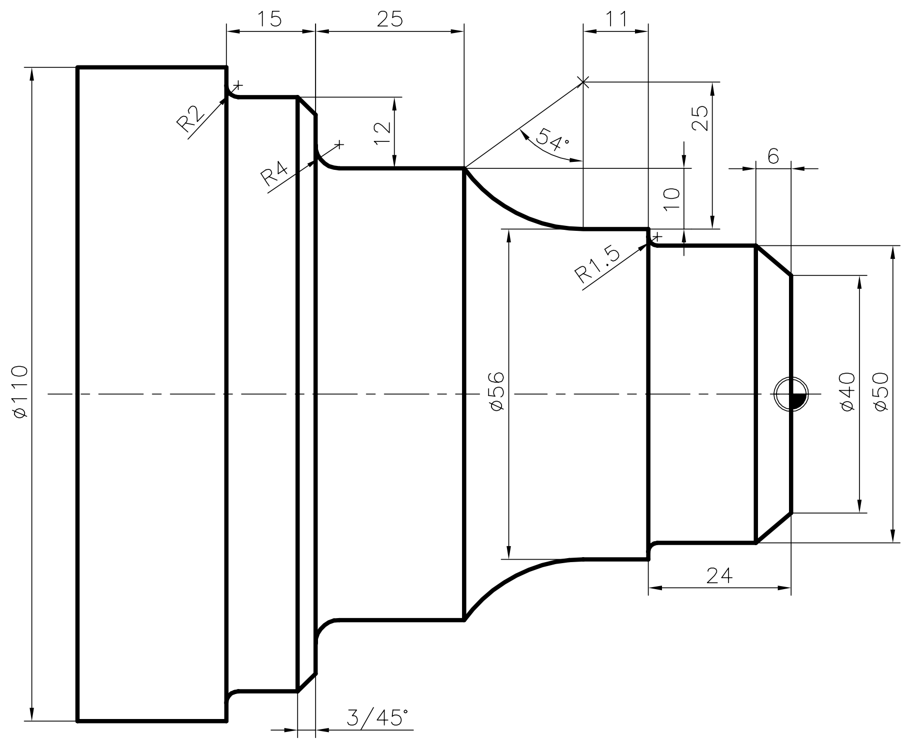
Rozwiązanie:

```
%_N_EX02_MPF
; 12-08-2003
N5 G54 G71 G90 G94 G17
N10 S800 F100 T1 D1 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 X60
N35 G2 I20 J0 AR=90
N40 G1 X110
N45 X130 Y54.641
N50 CT X125 Y85
N55 G91 G1 X-25
N60 G2 J20 AR=210
N65 G1 G90 Y135
N70 X=IC(-45)
N75 G2 X15 AR=180
N80 G1 X0 RND=8
N90 G1 Y70
N95 X10
N100 CIP Y=IC(-20) I1=79 J1=60
N105 G1 X0
N110 G3 X-5 Y15 I=AC(15) J-20
N115 G1 X0
N120 G2 J-15 AR=90
N125 G0 Z100
N130 G53 T0 D0 G0 X300 Y300 Z200 M9 M5 M30
```

4.2.5. Przykład dla obróbki tokarskiej

Zaprogramować ruch narzędzia po konturze przedstawionym na Rys. 77 w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**.

Ruch rozpocząć od punktu (0,0) w kierunku przeciwnym do ruchu wskazówek zegara. Obróbkę wykonać nożem zdzierakiem z płytką rombową 80°, o kącie przystawienia 95°, szerokości krawędzi 12 i promieniu naroża 1.2. Wykorzystać poznane funkcje definiowania współrzędnych.



Rys. 77. Szkic przedmiotu do przykładu obróbki konturu z interpolacją liniową i kołową

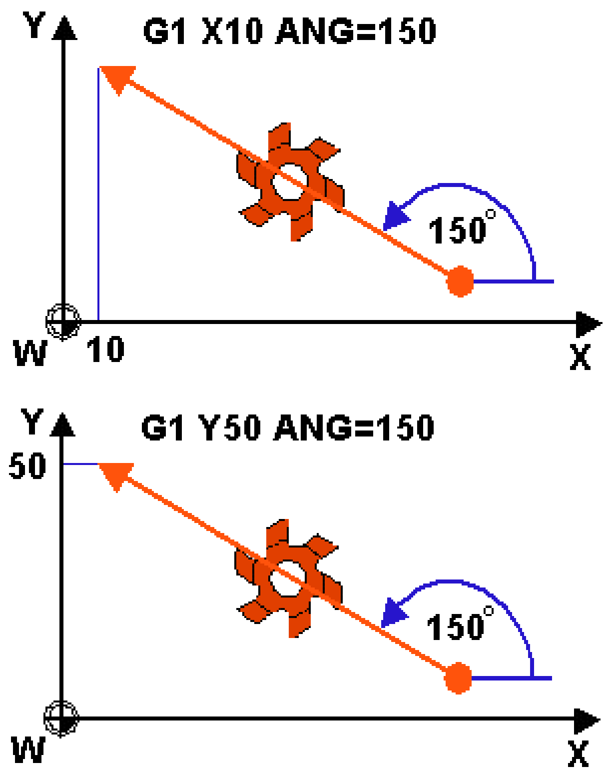
Rozwiązanie:

```
%_N_EX03_MPF
; 16-08-2004
N5 G40 G54 G71 G90 G96 DIAMON
N10 T1 D1 S150 F0.15 M4 M8
N15 G0 X0
N20 Z3
N25 G1 Z0
N30 X40
N35 Z-6 X50
N40 Z-24 RND=1.5
N45 X56
N50 G91 Z-11
N55 G2 I25 AR=54
N60 G1 Z-25 RND=4
N65 DIAMOF X=AC(12) CHR=3
N70 Z-15 RND=2
N75 G90 DIAMON X110
N80 G53 T0 D0 G0 Z300 X300 M9 M5
N85 M30
```

4.3. Programowanie z wykorzystaniem współrzędnych kątowych

Oprócz współrzędnych prostokątnych (kartezjańskich) dostępne są inne typy układów współrzędnych (biegunowe, walcowe, sferyczne itp.), z których układ sterowania sam przelicza współrzędne na standardowy układ kartezjański.

Najprostszym przykładem wykorzystania nieprostokątnego układu współrzędnych jest adres **ANG** (ang. *ANGle*) służący do programowania w interpolacji liniowej kąta, pod którym leży linia toru narzędzia (Rys. 78). Użycie tego adresu jest możliwe, jeżeli spełnione są dwa warunki: ruch odbywa się w aktualnej płaszczyźnie interpolacji (G17, G18, G19) oraz znany jest kąt prostej w stosunku do osi odciętych i jedna współrzędna (odcięta lub rzędna). Druga współrzędna jest obliczana przez układ sterowania. Kąt jest odmierzany w kierunku trygonometrycznym (dodatni dla zwrotu przeciwnego do ruchu wskazówek zegara, ujemny dla zwrotu zgodnego).



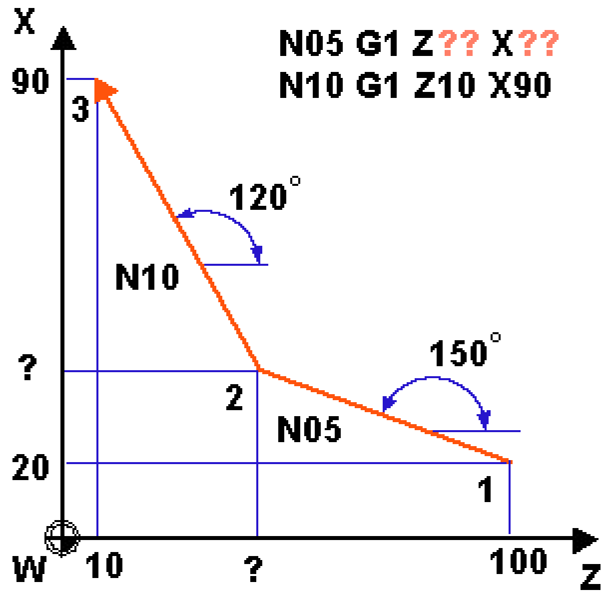
Rys. 78. Interpolacja liniowa z wykorzystaniem adresu **ANG**

Analizując wymiarowanie na rysunkach konstrukcyjnych, zwłaszcza przedmiotów osiowosymetrycznych, nierzadko można spotkać się z sytuacją, kiedy współrzędne liniowe jakiegoś punktu nie są podane, natomiast mogą zostać obliczone np. jako współrzędne punktu przecięcia się dwóch prostych pod zadanymi kątami, przechodzących przez punkty o danych współrzędnych – Rys. 79. Formalnie taki fragment konturu trzeba programować z wykorzystaniem dwóch bloków, przy czym należy obliczyć współrzędne punktu pośredniego (punktu przecięcia). Nic nie stoi na przeszkodzie aby wspomniane obliczenia dokonał układ sterowania – taki przypadek nazywa się programowaniem konturu przez dwa kąty. W tym celu należy posłużyć się dwoma adresami oznaczającymi kąt pochylenia linii prostej, przy warunkach

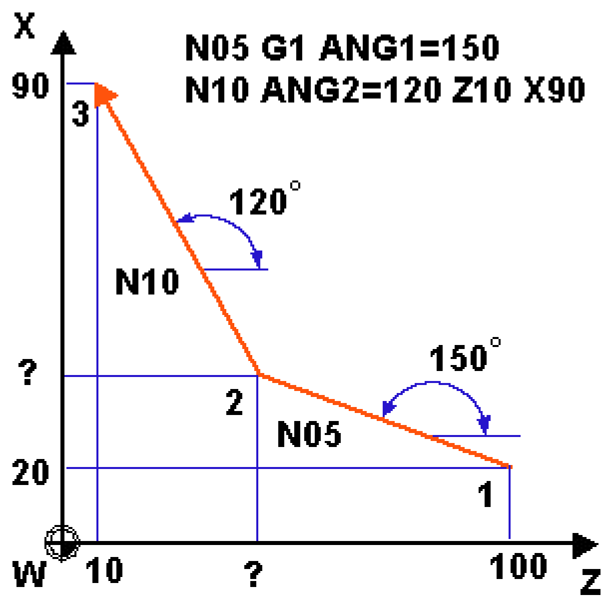
podobnych jak dla pojedynczego adresu ANG: **ANG1** i **ANG2**, programowanymi w osobnych blokach (Rys. 80):

G1 ANG1=...

G1 ANG2=... Z=Z_3 X=X_3



Rys. 79. Interpolacja liniowa – brak współrzędnych liniowych punktu pośredniego „2"



Rys. 80. Interpolacja liniowa przez dwa kąty

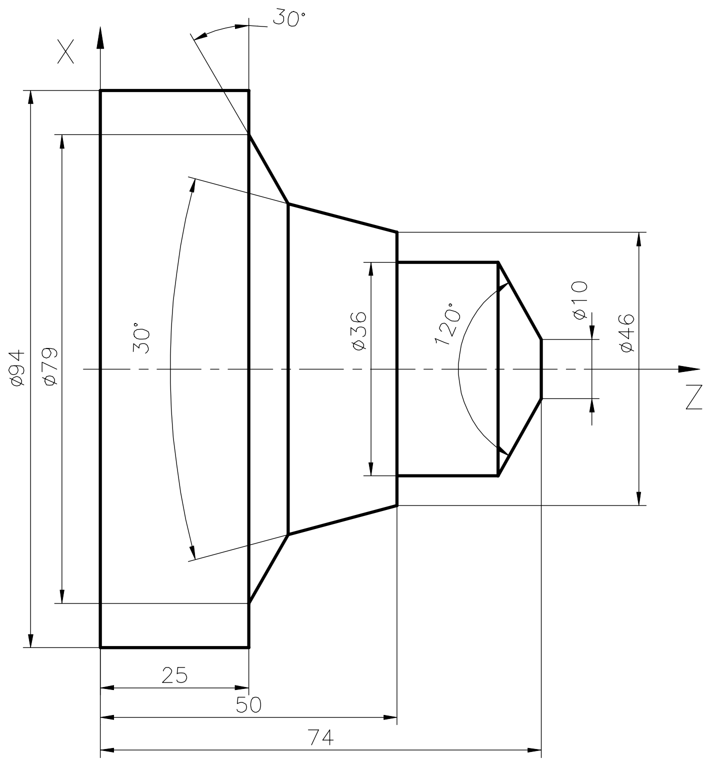
W punkcie pośrednim możliwe jest również wykonanie zaokrąglenia (RND) jak i sfazowania (CHR, CHF) konturu (rozdz. 3.5). Wtedy podobnie jak przy zwykłym zadawaniu współrzędnych pierwszy z bloków konturu przez dwa kąty dodatkowo musi zawierać adres dla zaokrąglenia/fazki:

G1 ANG1=... RND=.../CHR= .../CHF=...

G1 ANG2=... Z=Z_3 X=X_3

4.3.1. Przykład

Zaprogramować obróbkę wykańczającą powierzchni zewnętrznej wałka przedstawionego na Rys. 81. Rozpocząć od obróbki prawego czoła, następnie kolejne odcinki konturu. Wykorzystać funkcje programowania kąta dla interpolacji liniowej.



Rys. 81. Szkic przedmiotu do przykładu z programowaniem konturu przez dwa kąty

Rozwiązanie:

```
%_N_EX13_MPF
; 09-09-2003
N5 G40 G54 G71 G90 G95 DIAMON KONT G450
N10 T1 D1 S250 F0.1 M4 M8
N15 G0 X0
N20 Z77
N25 G1 Z74
N30 X10
N35 ANG=120 X36
N40 Z50
N45 X46
N50 ANG1=165
N55 ANG2=120 Z25 X79
N60 X94
N65 Z0
N70 G53 T0 D0 G0 Z300 X300 M9 M5
N75 M30
```

4.4. Programowanie we współrzędnych biegunowych i walcowych

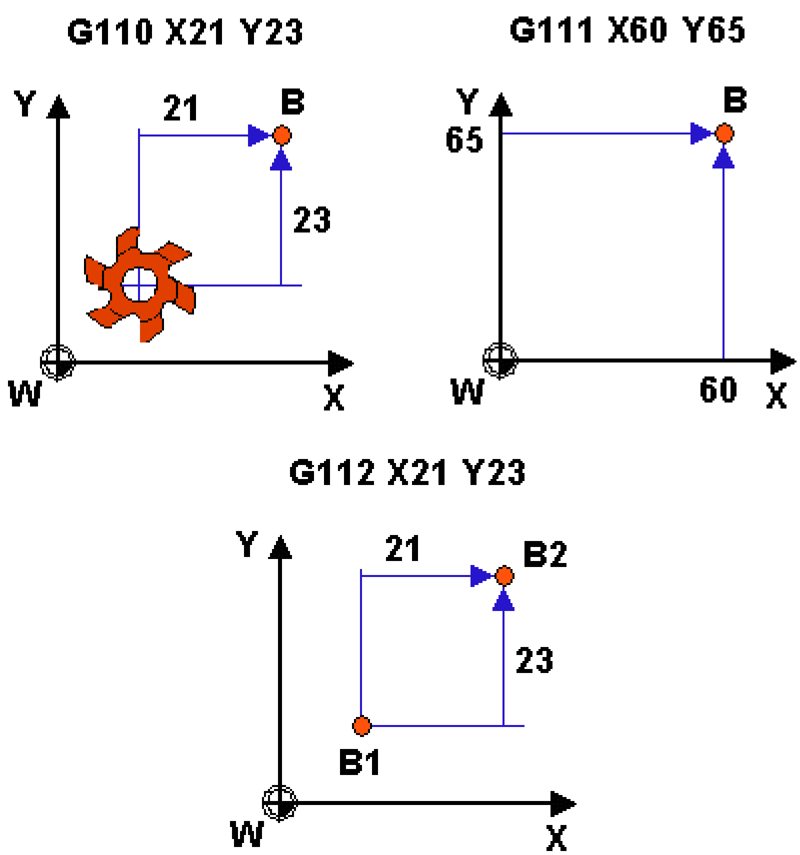
Drugim rodzajem układu współrzędnych, wykorzystującym wymiary kątowe, jest układ biegunowy (również walcowy). Ten rodzaj wyrażania współrzędnych wymaga określenia w pierwszej kolejności położenia bieguna (względem którego określa się współrzędne biegunowe). Do tego celu służą następujące niemodalne funkcje przygotowawcze (Rys. 82):

G110 – programowanie bieguna względem ostatniego położenia narzędzia (przyrostowo, niezależnie od funkcji G90/G91).

G111 – programowanie absolutne położenia bieguna (niezależnie od funkcji G90/G91).

G112 – programowanie bieguna względem ostatniego położenia bieguna (przyrostowo, niezależnie od funkcji G90/G91).

Ze względu na konieczność użycia adresów X, Y, Z wymienione powyżej adresy nie wolno łączyć w jednym bloku z żadnymi innymi funkcjami przygotowawczymi, a w szczególności z funkcjami interpolacji. Należy pamiętać również, że domyślne położenie bieguna to punkt zerowy aktualnego układu współrzędnych. Blok, w którym programuje się biegun nie wywołuje żadnego ruchu narzędzia, wartości współrzędnych użytych w poprzednich blokach są przywracane w bloku następnym (możliwość programowania przyrostowego).



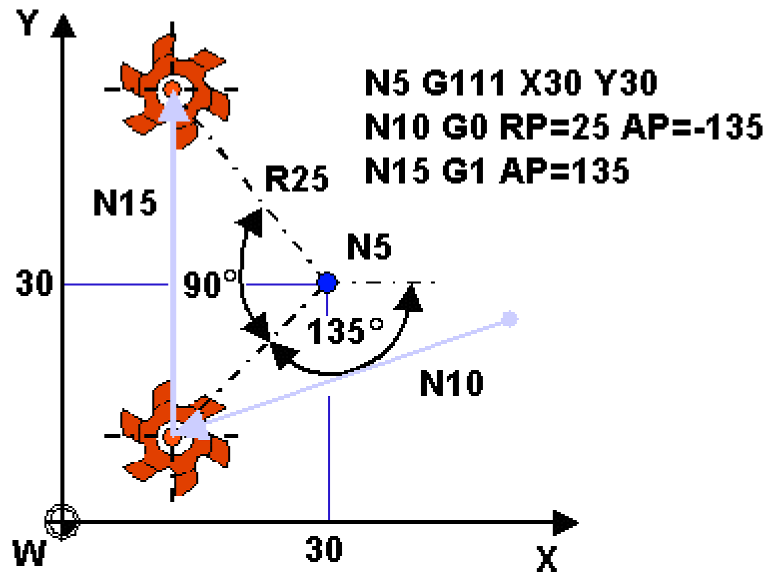
Rys. 82. Programowanie położenia bieguna dla biegunowego układu współrzędnych

Współrzędne biegunowe programuje następującymi adresami (Rys. 83):

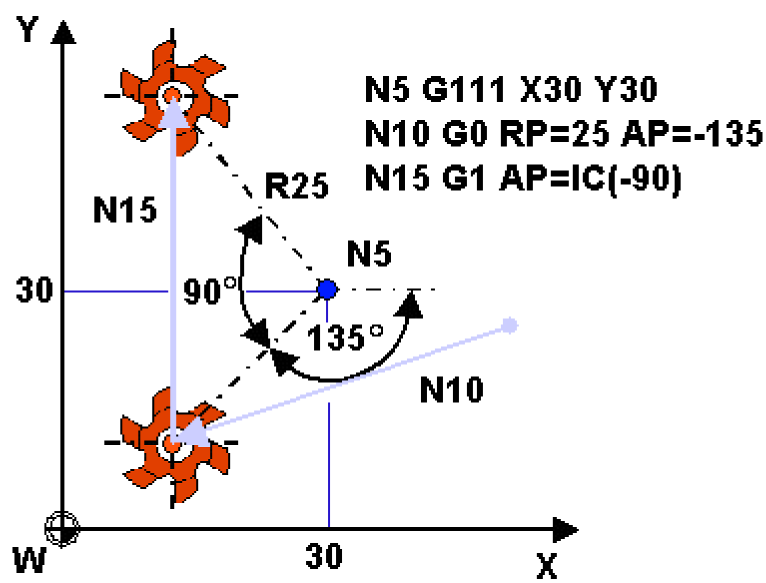
RP – promień (ang. *Radius Polar*);

AP – kąt wodzący (ang. *Angle Polar*);

przy czym adresy te są modalne, o domyślnych wartościach równych 0. Mogą być programowane absolutnie (domyślnie) lub przyrostowo (za pomocą funkcji niemodalnej **IC** – Rys. 84). **Programowanie współrzędnych biegunowych zawsze dotyczy aktualnej płaszczyzny układu współrzędnej**, programowanej adresem G17/G18/G19. Jeżeli użyje się współrzędnej osi dosuwowej to mamy do czynienia ze współrzędnymi walcowymi. Należy pamiętać, że **nie można łączyć współrzędnych z różnych rodzajów układów**, np. prostokątnego i biegunowego, można natomiast w sposób dowolny używać ich w programie sterującym, przeplatając bloki z różnymi rodzajami współrzędnych.



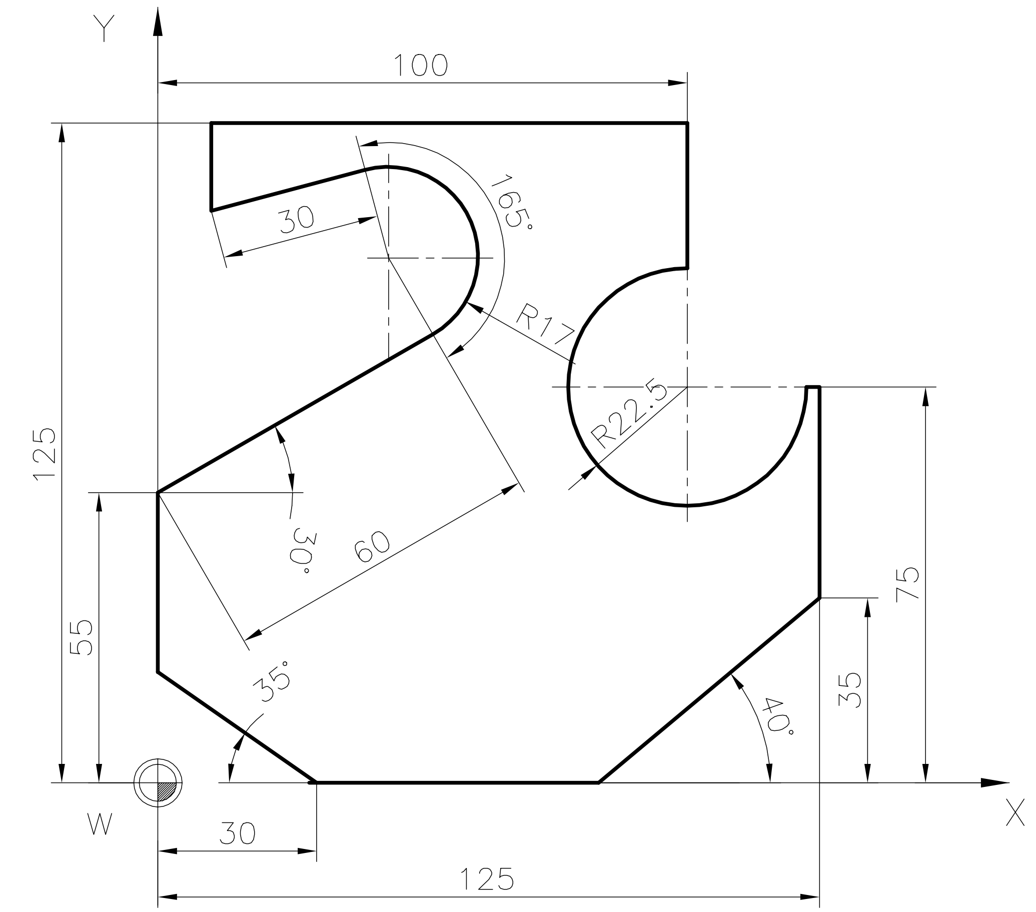
Rys. 83. Programowanie współrzędnych w układzie biegunowym



Rys. 84. Przyrostowe programowanie kąta we współrzędnych biegunowych

4.4.1. Przykład

Zaprogramować ruch narzędzia po konturze przedstawionym na Rys. 85 w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**. Ruch rozpocząć od punktu (0,0) w kierunku zgodnym do ruchu wskazówek zegara. Przyjąć poziom materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 16 mm.



Rys. 85. Szkic przedmiotu do przykładu obróbki konturu z wykorzystaniem współrzędnych biegunowych

Rozwiązanie:

```
%_N_EX04_MPF
; 17-08-2003
N5 G40 G54 G71 G90 G94
N10 T1 D1 S1500 F250 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 Y55
N35 G111 X0 Y55
N40 AP=30 RP=60
N45 G110 X-8.5 Y14.722
N50 CT AP=105 RP=17
N55 G110 X0 Y0
N60 G1 AP=195 RP=30
N65 Y125
```

```
N70 X100
N75 G111 X100 Y75
N80 AP=90 RP=22.5
N85 G3 J-20 AR=270
N90 G1 X125
N95 Y35
N100 ANG=220 Y0
N105 X30
N110 ANG=145 X0
N115 G0 Z100
N120 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N125 M30
```

4.5. Transformacje układów współrzędnych (FRAMES)

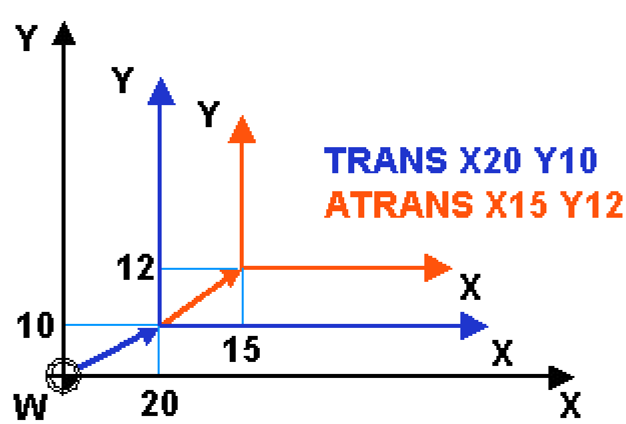
Idea programowalnych transformacji układów współrzędnych (FRAMES) polega na definiowaniu reguł przekształcania jednego układu współrzędnych w drugi poprzez zastosowanie przesunięć, obrotów itp. transformacji geometrycznych. Reguły te są zapisywane w postaci macierzy, gdyż proces przeliczania współrzędnych z jednego układu w drugi najprościej zrealizować przy pomocy tego mechanizmu.

W układzie sterowania Sinumerik 840D zawarte są cztery podstawowe transformacje, programowane przy użyciu ośmiu niemodalnych funkcji przygotowawczych. Podzielone są one na dwie grupy:

- ➔ Funkcje działające w odniesieniu do bieżącego ustawczego układu współrzędnych (G54, G55, ...): TRANS, ROT, MIRROR, SCALE;
- ➔ Funkcje działające addytywnie (ang. *Additive*) w odniesieniu do bieżącego układu współrzędnych: ATRANS, AROT, AMIRROR, ASCALE. Działanie tych funkcji jest kasowane przez przywołanie funkcji TRANS, ROT, MIRROR, SCALE, lub przez przywołanie ustawczego układu współrzędnych (G54, G55, ...).

Funkcje te muszą być programowane osobno w oddzielnych blokach z uwagi na konieczność współpracy z adresami geometrycznymi (np. X, Y, Z). Ich znaczenie jest następujące:

TRANS, ATRANS – (ang. *TRANSlation*) przesunięcie (translacja) początku układu współrzędnych o zadany wektor, którego współrzędne są programowane pod adresami X, Y, Z w bieżącym układzie współrzędnych (Rys. 86).

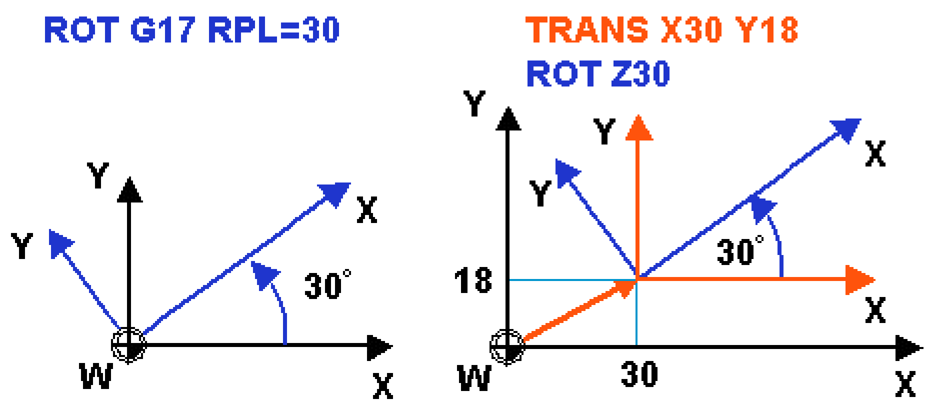


Rys. 86. Przesunięcie układu współrzędnych (TRANS, ATRANS)

ROT, AROT – (ang. ROTation) obrót układu współrzędnych wokół zadanej osi o dany kąt (Rys. 87). Kąt ten może być zadany dwojako:

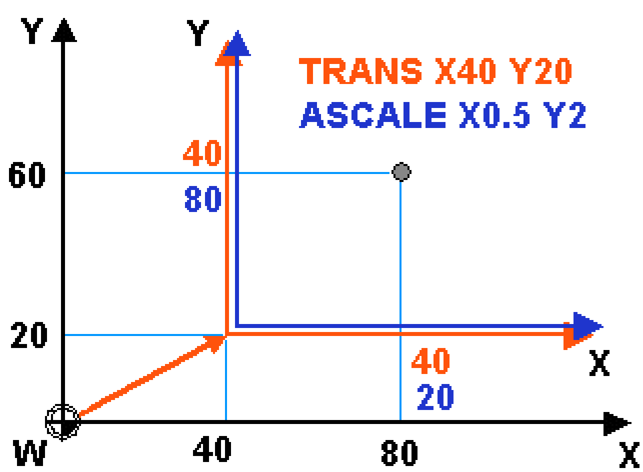
- ➔ **RPL** – (ang. *Rotation PLane*) obrót wokół osi dosuwowej, przy czym programowany jest adres G17/G18/G19 określający płaszczyznę w której ten obrót się odbywa (tylko jeden obrót w bloku);
- ➔ **X, Y, Z** – obroty wokół geometrycznych (maksimum trzy obroty w jednym bloku) – wartości adresów programują kąt obrotu wokół danej, przy czym jest zawsze zachowana następująca kolejność obliczania transformacji: wokół osi Z, Y i X niezależnie od kolejności adresów w bloku.

Kierunek dodatni kąta obrotu jest przeciwny do ruchu wskazówek zegara (kierunek geometryczny dodatni).



Rys. 87. Obrót układu współrzędnych wokół osi (ROT, AROT)

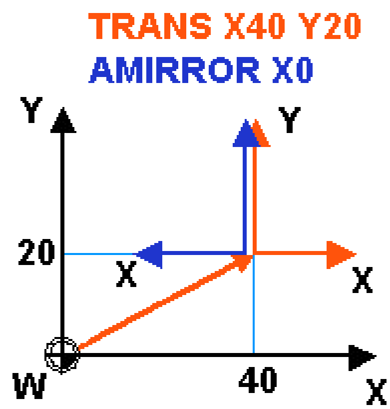
SCALE, ASCALE – zmiana współczynnika skali osi układu współrzędnych (Rys. 88). Programować można współczynniki skali osobno dla każdej osi pod adresami X, Y i Z. Przy obliczaniu współrzędnych w układzie bazowym wartości współrzędnych w układzie poddanym skalowaniu są dzielone przez zaprogramowane współczynniki skalujące.



Rys. 88. Skalowanie osi układu współrzędnych (SCALE, ASCALE)

MIRROR, AMIRROR – symetria osiowa (odbicie lustrzane) układu współrzędnych (Rys. 89). Programowana jest oś, która podlega transformacji

przez podanie w bloku adresu X, Y lub Z, przy czym wartość tych adresów jest dowolna (nie wpływa na transformację). Funkcje MIRROR i AMIRROR automatycznie zmieniają kierunki interpolacji kołowej (G2, G3) oraz kierunki kompensacji promienia narzędzia (G41, G42).



Rys. 89. Symetria osiowa (odbicie lustrzane) układu współrzędnych (MIRROR, AMIRROR)

Jeżeli w bloku występują same adresy TRANS, ROT, SCALE lub MIRROR bez parametrów definiujących transformacje, to powoduje to kasowanie wszystkich programowalnych zmian układu współrzędnych i powrót do nieprzekształconego, ustawczego układu współrzędnych (G54, G55, ...).

4.5.1. Przykład

Zaprogramować ruch narzędzia po czterech elementach konturu przedstawionych na Rys. 90 w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**. Ruch rozpocząć od punktu lewego dolnego narożnika w kierunku zgodym do ruchu wskazówek zegara. Przyjąć poziom materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 6 mm. Wykorzystać podprogram obróbki podanego kształtu (patrz rozdz. 6).

Rozwiązanie:

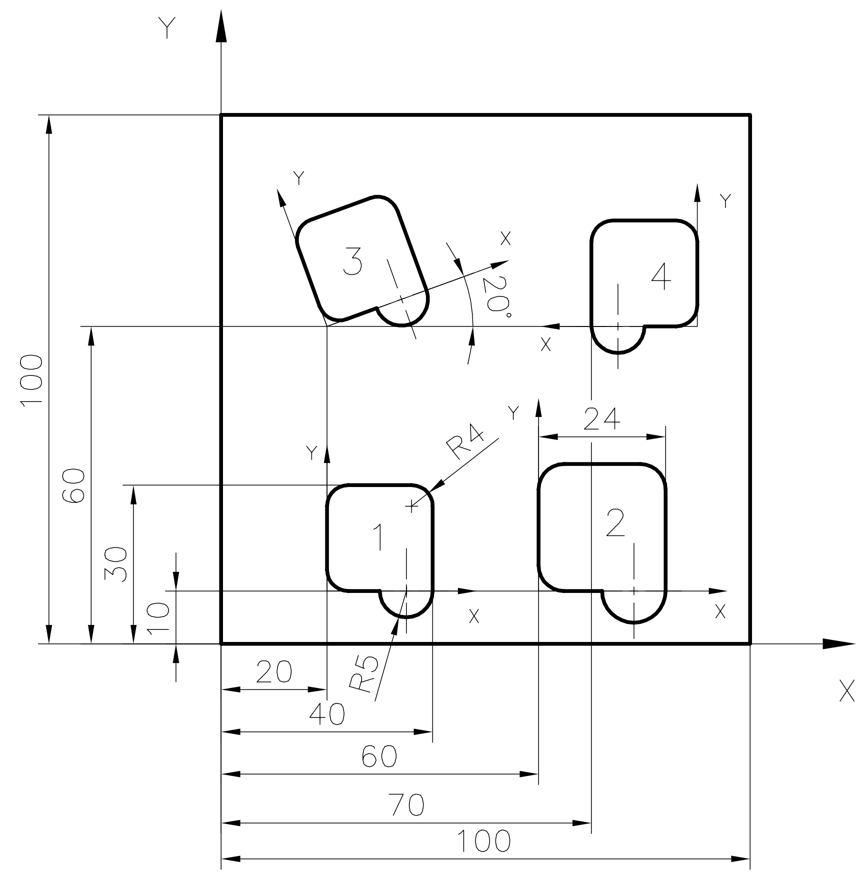
Program główny:

```
%_N_EX05_MPF
; 18-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S1000 F100 M3 M8 M6
; ELEMENT 1
N15 TRANS X20 Y10
N20 L5 P1
; ELEMENT 2
N25 TRANS X60 Y10
N30 ASCALE X1.2 Y1.2
N30 L5 P1
; ELEMENT 3
N35 TRANS X20 Y60
N40 AROT RPL=20
N45 L5 P1
; ELEMENT 4
N50 TRANS X90 Y60
```

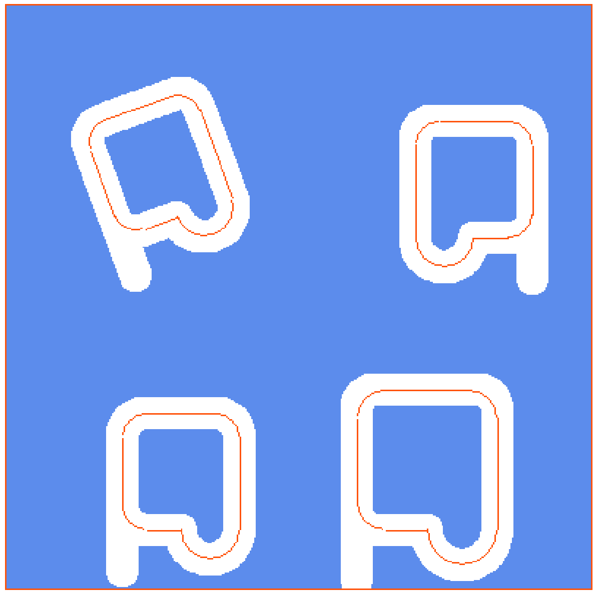
```
N55 AMIRROR X0
N60 L5 P1
N65 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N70 M30
```

Podprogram obróbki pojedynczego konturu przy przyjęciu lokalnego układu współrzędnych:

```
%_N_EX05_SPF
; PODPROGRAM KONTURU DO EX05
N5 G0 X0 Y-7
N10 G1 Z-5
N15 G1 Y20 RND=4
N20 X20 RND=4
N25 Y0
N30 G2 I-5 AR=180
N35 G1 X0 RND=4
N40 Y5
N45 G0 Z5
N50 M17
```



Rys. 90 Szkic przedmiotu do przykładu obróbki konturu z wykorzystaniem transformacji układu współrzędnych



Rys. 91 Wynik symulacji programu sterującego EX05

5. NARZĘDZIA – WYMIARY, PARAMETRY PRACY, KOMPENSACJA PROMIENIA

5.1. Rejestry narzędziowe

Jak wspomniano wcześniej (rozdz. 1.7), układ sterowania dla prawidłowego funckjonowania musi znać wymiary charakterystyczne narzędzia (nazywane korekcyjnymi). Są one przechowywane w **rejestrach narzędziowych**. Mają one postać adresowanych rekordów, zawierających pewną liczbę pól o wartościach rzeczywistych, przechowujących dane narzędziowe. Część z nich wykorzystywana jest bezpośrednio przez układ sterowania, pozostałe mogą być traktowane w sposób dowolny. Dla przykładu w rejestrach narzędziowych mogą być zawarte informacje o planowanym i aktualnym czasie pracy ostrza narzędzia, liczbie obrobionych sztuk przedmiotów itp.

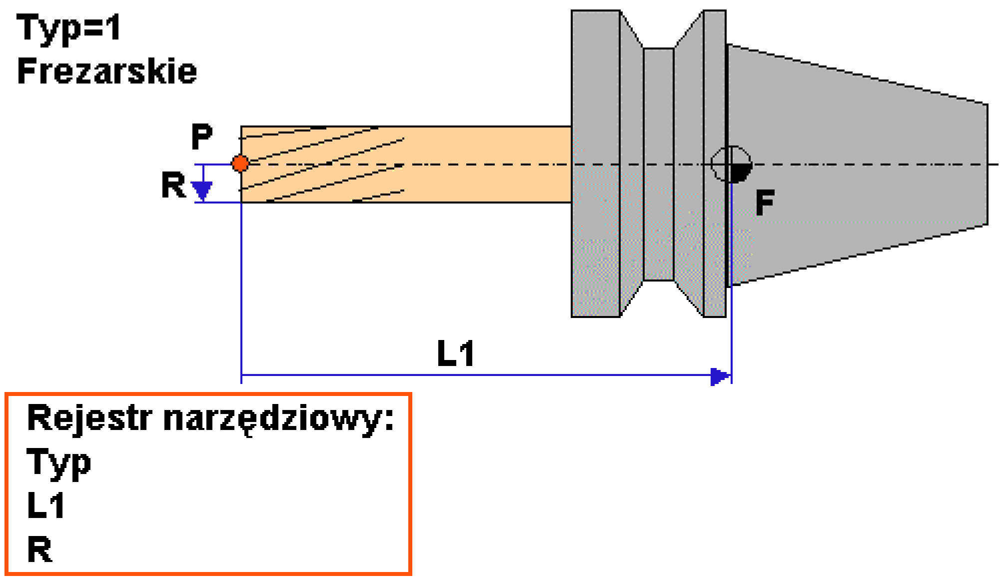
Sposób adresowania rejestrów może być różny. W wielu układach sterowania jest to ciągły układ od D1, D2, D3, itd., gdzie numery rejestrów są niezależne od numerów aktualnych narzędzi (adres T). W układzie sterowania Sinumerik 810D/840D przewidziano po oddzielnych rejestrów dla każdego narzędzia, adresowanych jako D1, D2,..., D9. Jednoznaczne zidentyfikowanie rejestru narzędziowego wymaga zaprogramowania zarówno numeru narzędzia (adres T), jak i przypisanego do niego rejestru (adres D). Zatem rejestry T1 D1 i T2 D1 to **dwa różne rejestry**!

Istnieje specjalny, niedostępny do edycji rejestr D0, traktowany jako zawierający zerowe wymiary narzędzia, co prowadzi do bezpośredniego programowania ruchu punktu kodowego **F**. Swobodnie można natomiast zmieniać zawartość pozostałych rejestrów. Każdy z pozostałych rejestrów zawiera max. 25 wartości numerycznych, przy czym zazwyczaj tylko część z nich jest używana do opisu parametrów narzędzi przy kompensacji ich długości i promienia ostrza.

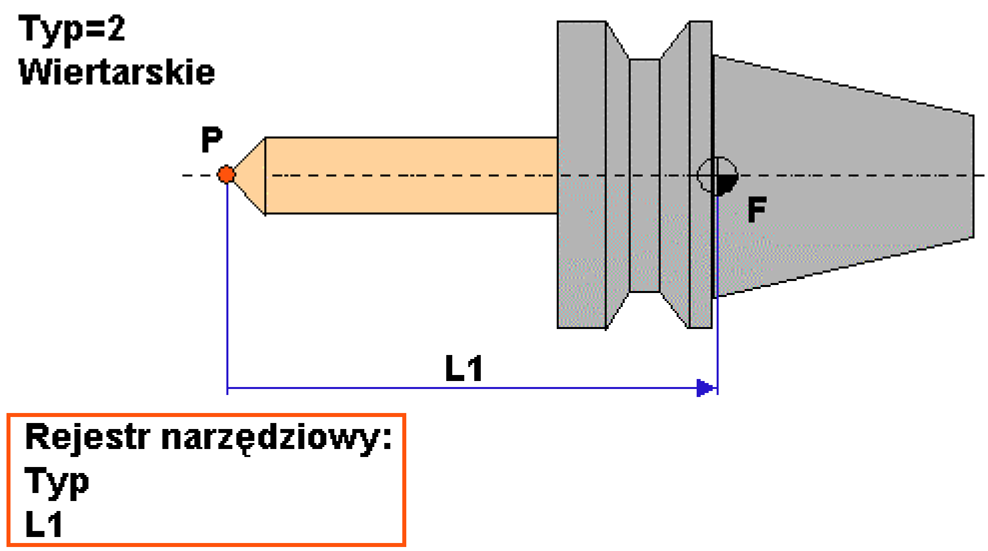
Najważniejszym parametrem w rejestrze narzędziowym jest **typ narzędzia**, określający przynależność narzędzia do grupy narzędzi. Obecnie zdefiniowane grupy narzędzi to:

➔ 1xx – narzędzia frezarskie;
➔ 2xx – narzędzia wiertarskie;
➔ 4xx – narzędzia szlifierskie;
➔ 5xx – narzędzia tokarskie;
➔ 7xx – narzędzia do rowków.

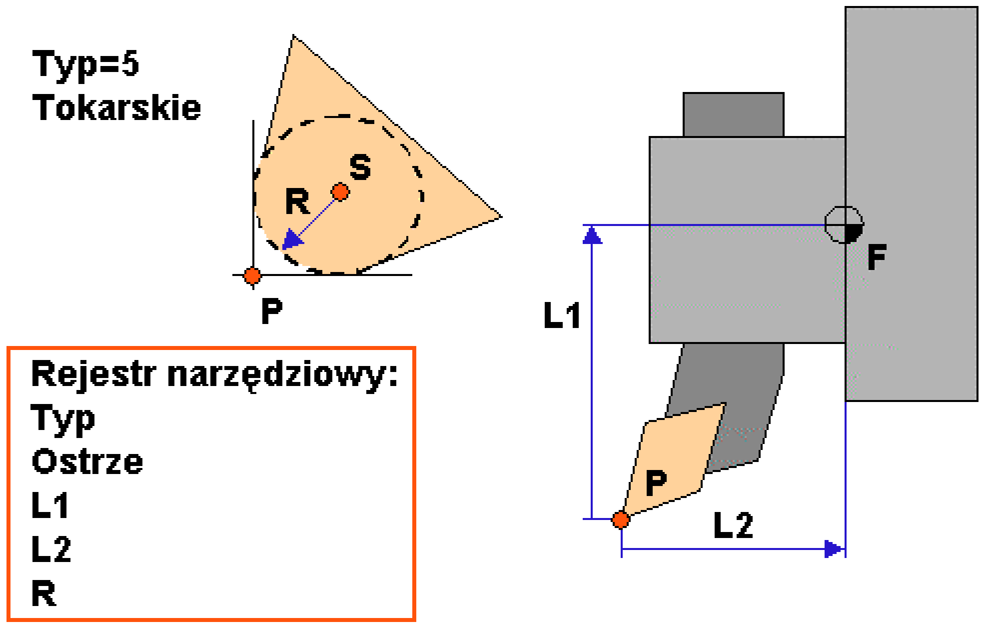
"xx" w powyższych oznaczenia zastępuje wartości liczbowe, odpowiadające konkretnemu rodzajowi narzędzia w ramach typu, np. 500 opisuje nóż zdzierak, 250 rozwiertak itd. W zależności od typu narzędzia zmienia się zawartość i interpretacja rejestrów narzędziowych. Poniżej przedstawiono najważniejsze dane dla najczęściej używanych typów narzędzi (Rys. 92, Rys. 93 i Rys. 94).



Rys. 92. Najważniejsze parametry narzędzi frezarskich

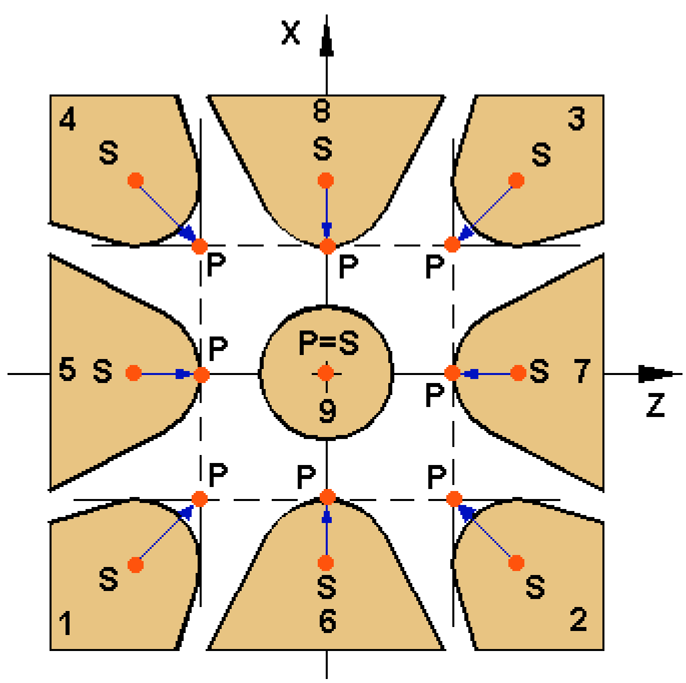


Rys. 93. Najważniejsze parametry narzędzi wiertarskich



Rys. 94. Najważniejsze parametry narzędzi tokarskich

Szczególnym typem narzędzi jest grupa “5xx” (narzędzia tokarskie). Jako jedyna posiada w rejestrze narzędziowym pozycję określającą położenie ostrza względem punktu kodowego **P** (Rys. 95). Definiuje ono kierunek punktu kodowego **P** narzędzia (na przecięciu się stycznych do krawędzi narzędzia – wynika to z zasady pomiarów narzędzia) z punktu **S** (środek okręgu wpisanego w naroże narzędzia). Np. położenie ostrza kodowane jako **3** oznacza, że punkt **P** jest przesunięty w obu osiach (Z i X) o wartość promienia naroża w kierunku ujemnym. Zasadność tej operacji zostanie omówiona w rozdz. 2.3 poświęconym kompensacji promienia narzędzia.



Rys. 95. Definicja położenia ostrza dla narzędzi tokarskich (parametr **Ostrze** na Rys. 94)

5.2. Parametry pracy narzędzi

Do parametrów technologicznych, związanych z pracą narzędzi, należą posuw (adres F) i prędkość skrawania (adres S). Mogą one być wyrażone w różny sposób i w różnych jednostkach, np. posuw może być programowany w [mm/min] lub [cal/min]. W tym przypadku nie mają zastosowania funkcje G70/G71. Zasadniczo jednostka posuwu jest określana na podstawie zmiennych systemowych układu sterowania. Jako uzupełnienie wspomnianych dwóch funkcji przygotowawczych w Sinumeriku zaimplementowano jeszcze dwie funkcje należące do tej samej grupy, określające wymiar stosowanych jednostek. Są to:

G700 – jednostki dodatkowe dla wymiarów geometrycznych [cal] i posuwu [cal/min];

G710 – jednostki podstawowe dla wymiarów geometrycznych [mm] i posuwu [mm/min].

W Sinumeriku, choć w innych układach sterowania jest to czasami inaczej, do programowania rodzaju i jednostek parametrów technologicznych służą funkcje przygotowawcze, należące do **jednej grupy**.

Do programowania posuwu służą następujące funkcje:

G93 – odwrotność czasu trwania bloku **F** [1/s];

G94 – posuw minutowy **F** [mm/min];

G95 – posuw obrotowy **F** [mm/obr].

W praktyce posuw minutowy (G94) jest używany na frezarkach bądź przy frezowaniu na centrach tokarsko-frezarskich, posuw obrotowy (G95) prawie wyłącznie przy toczeniu. Funkcja G93 jest rzadko stosowana, ponieważ charakteryzuje się zmienny posuwem, zależnym od drogi programowanej w jednym bloku.

Do programowania prędkości skrawania/prędkości obrotowej służą następujące funkcje:

G96 – włączenie stałej prędkości skrawania **S** [m/min];

G961 – włączenie stałej prędkości skrawania **S** [m/min];

G97 – wyłączenie stałej prędkości skrawania **S** [obr/min];

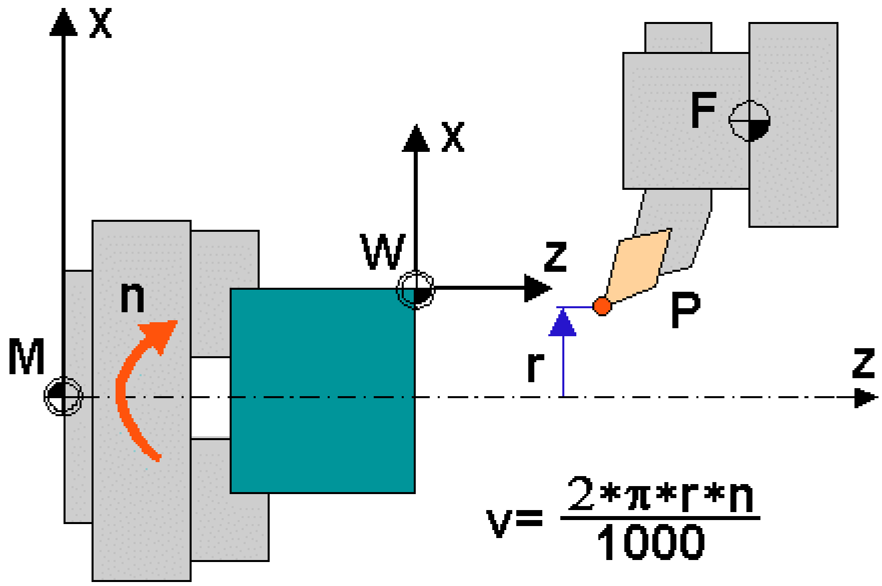
G971 – wyłączenie stałej prędkości skrawania **S** [obr/min].

Domyślnym rodzajem pracy jest **stała prędkość obrotowa wrzeciona *n*** – **S** wyrażone w [obr/min] (podobnie jak ma to miejsce dla obrabiarek konwencjonalnych). Włączanie i wyłączanie **stałej prędkości skrawania *v*** (zmienna wartość prędkości obrotowej wrzeciona) jest w praktyce używane na tokarkach. Zależność pomiędzy obu prędkościami jest ogólnie znana:

$$v = \frac{2 \cdot \pi \cdot r \cdot n}{1000} \qquad (\mathrm{G97, G971})$$
$$n = \frac{1000 \cdot v}{2 \cdot \pi \cdot r} \qquad (\mathrm{G96, G961}) \tag{8}$$

Wartość promienia toczenia r w powyższym wzorze jest równa odległości punktu kodowego **P** narzędzia od osi wrzeciona, mierzona w układzie maszynowym MKS – Rys. 96. Stała prędkość obrotowa jest używana przede wszystkim na frezarkach. Na tokarkach jest używana przy takich zabiegach jak wiercenie osiowe, toczenie gwintu,

przecinanie, toczenie rowków. Stała prędkość skrawania na tokarkach jest stosowana przy toczeniu i wytaczaniu w celu uzyskania dobrej jakości powierzchni



Rys. 96. Zależność między prędkością obrotową a stałą prędkością skrawania

We wzorze (8) na prędkość obrotową wrzeciona *n* przy stałej prędkości skrawania *v* w mianowniku występuje wartość promienia *r*. Jeżeli narzędzie zbliża się do osi to maleje *r* i rosną obroty *n* (zjawisko to jest nazywane rozbieganiem wrzeciona), teoretycznie do nieskończoności. Stan ten może być niebezpieczny (np. ze względu na zastosowany uchwyt). Dlatego konieczne jest ograniczanie obrotów wrzeciona. Maksymalne obroty wrzeciona ograniczone są przez:

1. Maksymalne obroty silnika napędzającego wrzeciono;
2. Wartość maksymalną i minimalną obrotów wrzeciona, zadawanych z pulpitu operatorskiego (zalecane) lub programowo przez funkcje przygotowawcze:
   G25 – minimalne obroty wrzeciona (programowane pod adresem **S**);
   G26 – maksymalne obroty wrzeciona (programowane pod adresem **S**).
   Przykład:
   N05 G25 S100
   N10 G25 S3000
   Powyższe bloki ograniczają obroty wrzeciona w zakresie 100÷3000 [obr/min].
3. Maksymalne obroty wrzeciona przy stałej prędkości skrawania (G96), programowane pod adresem **LIMS** (ang. *LIMit Speed*).
   Przykład:
   N10 LIMS=2000
   Ograniczenie obrotów przy stałej prędkości skrawania do 2000 [obr/min].

Należy pamiętać, że maksymalne obroty wrzeciona, programowane przez adres LIMS nie mogą być większe niż ustawione przez operatora. Dlatego w programie należy **unikać ograniczania obrotów przez adres G26**. Przy wyłączeniu stałej prędkości skrawania funkcją G97 przy włączonych obrotach wrzeciona **nie trzeba zadawać nowej wartości adresu S** jako stałej prędkości obrotowej, gdyż wtedy zostaje **domyślnie przyjęta aktualna prędkość obrotowa**.

Ponieważ funkcje w omawianej grupie wpływają zarówno na interpretację posuwu jak prędkości skrawania/obrotowej obowiązują tu pewne powiązania ich aktywności. Są one przedstawione w Tabl. 1.

Tabl. 1. Powiązania funkcji przygotowawczych w grupie technologicznej

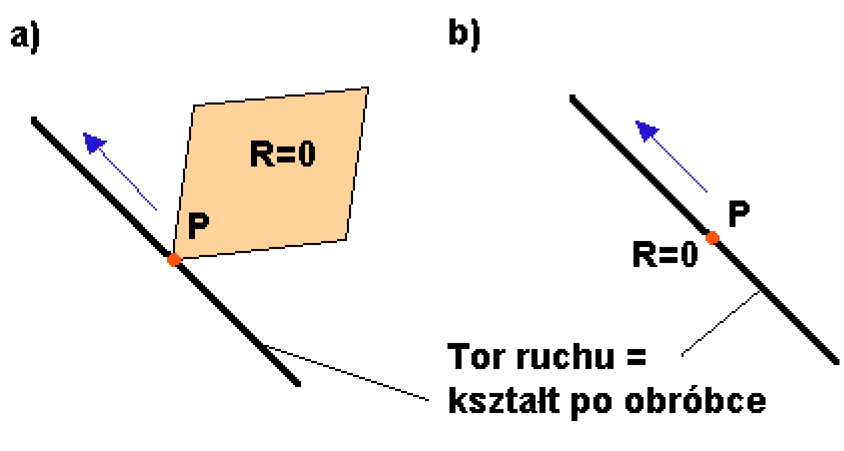
Funkcja aktywna	Stan jak przy funkcji	**S**	**F**
G93	G97	[obr/min]	[1/s]
G94	G97	[obr/min]	[mm/min]
G95	G97	[obr/min]	[mm/obr]
G96	G95	[m/min]	[mm/obr]
G961	G94	[m/min]	[mm/min]
G97	G95	[obr/min]	[mm/obr]
G971	G94	[obr/min]	[mm/min]

5.3. Kompensacja promienia narzędzia

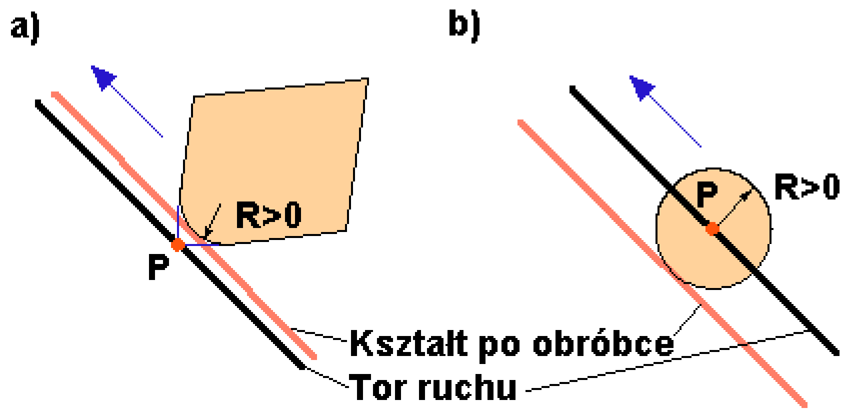
5.3.1. Istota kompensacji promienia narzędzia

Obróbka skrawaniem ma na celu uzyskanie żądanego kształtu i wymiarów przedmiotu obrabianego. Są one wynikiem wzajemnego ruchu przedmiotu i narzędzia, a właściwie jego krawędzi skrawających. Ze względów praktycznych punkt kodowy narzędzia na ogół nie pokrywa się z punktem (punktami) kształtującymi przedmiot obrabiany. Stąd kształt powstały po obróbce nie jest zgodny z kształtem opisanym w programie sterującym – powstaje pewien błąd obróbki (patrz Rys. 50). Aby wyeliminować ten błąd należy dokonać takiej korekty toru ruchu punktu kodowego narzędzia, aby w efekcie uzyskać zamierzony kształt i wymiary przedmiotu obrabianego.

Wspomniany błąd nie powstałby gdyby obróbka odbywała się narzędziami o idealnych kształtach (np. noży tokarskich) lub o zerowej średnicy (np. frezów) – Rys. 97. W rzeczywistości obróbka odbywa się narzędziami o określonej średnicy czy rzeczywistym kształcie – Rys. 98. Ponieważ w obu przypadkach decydujące znaczenie ma wartość promienia (naroża ostrza, samego narzędzia) stąd procedurę korekcyjną nazwano **kompensacją promienia narzędzia** (można też spotkać określenie **korekcja promienia narzędzia**).



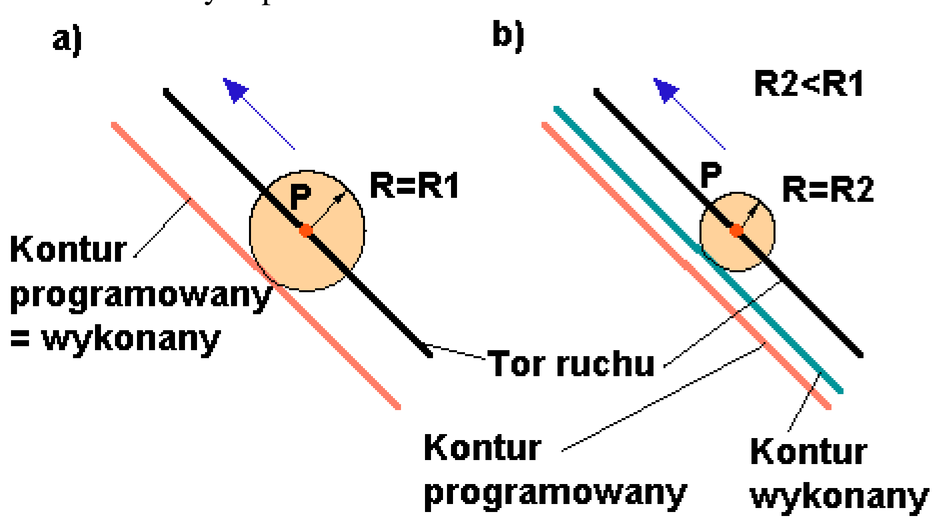
Rys. 97. Kształt po obróbce narzędziem o idealnym kształcie (a) lub zerowej średnicy (b)



Rys. 98. Kształt po obróbce narzędziem o rzeczywistym kształcie (a) lub niezerowej średnicy (b)

W praktyce programowania stosuje się dwie metody kompensacji promienia narzędzia:

1. Programową – zmodyfikowany tor ruchu narzędzia, uwzględniający rzeczywistą wartość promienia, tworzony jest w sposób ręczny (dawniej) lub generowany automatycznie przez systemy CAM (obecnie). Zaletą tego rozwiązania jest możliwość pełnej weryfikacji programu przed jego wczytaniem do układu sterowania, wadą zaś konieczność zmiany programu po zmianie promienia narzędzia (np. wynikającej ze zużycia) – Rys. 99. Tym niemniej takie rozwiązanie jest stosowane, przede wszystkim przy obróbce powierzchni swobodnych (np. powierzchni NURBS), wymagających kompensacji trójosiowej (przestrzennej) i wielu skomplikowanych, czasochłonnych obliczeń, które są wykonywane przez specjalizowane moduły systemów CAM. Minimalne zmiany wymiarów narzędzia nie mają w takim przypadku tak dużego znaczenia z uwagi na większe tolerancje wymiarowe obrabianych powierzchni.

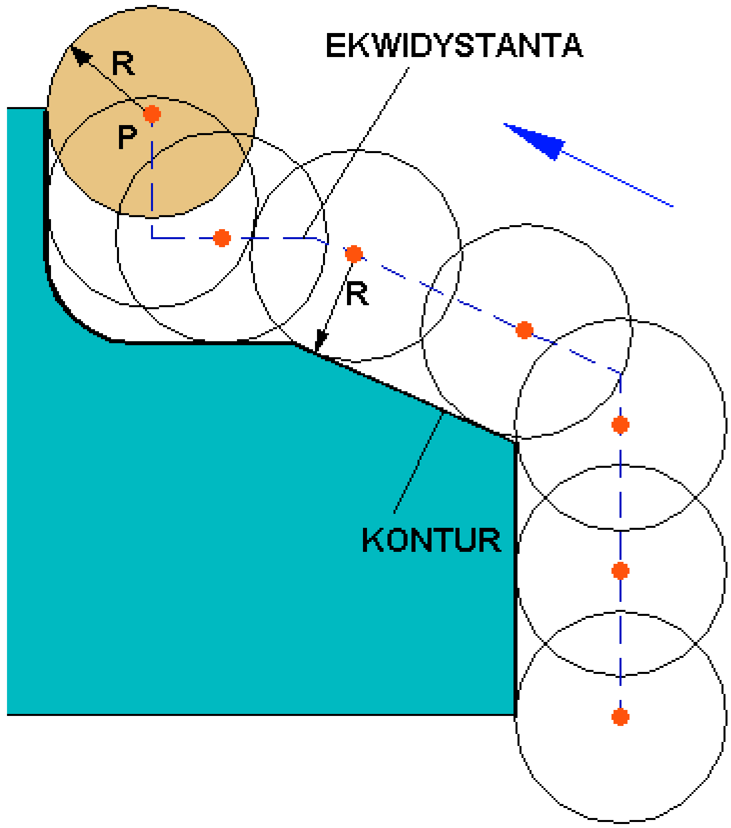


Rys. 99. Zmiana konturu obrobionego na wskutek zmian promienia narzędzia przy kompensacji programowej

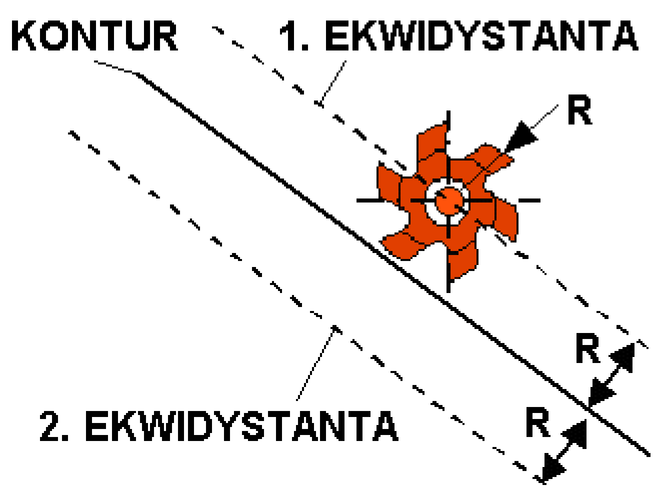
2. Automatyczną – wykonywaną przez układ sterowania przy każdym wykonaniu programu na podstawie aktualnej wartości promienia narzędzia, pobranej

z rejestrów narzędziowych. Zaletą takiego podejścia jest dokładne i precyzyjne uwzględnienie najmniejszych zmian wymiarów narzędzia (a co za tym idzie osiąga się dużą dokładność obróbki) oraz łatwość programowania – programuje się kontur nominalny, taki jaki ma powstać po obróbce, a więc zwymiarowany na rysunku konstrukcyjnym. Wadą zaś konieczność każdorazowego obliczania rzeczywistego toru narzędzia przy wykonaniu programu. Z uwagi jednak na dużą moc obliczeniową dzisiejszych układów sterowania wada ta nie ma istotnego wpływu na szybkość realizacji programu i samej obróbki.

Zarówno kompensacja programowa jak i automatyczna są oparte na tej samej zasadzie funkcjonowania, która omówiona zostanie na przykładzie obróbki frezowania frezem palcowym. Pracujące swoim obwodem narzędzie kształtuje obrabiany kontur różnymi punktami na tymże obwodzie (Rys. 100), jednak ich wspólną cechą jest stała odległość od punktu kodowego **P**, równa promieniowi narzędzia. Tor punktu kodowego **P** jest zatem zawsze w stałej odległości od programowanego konturu. Z tego względu zakreśla krzywą równoległą do konturu, nazywaną **ekwidystantą** (krzywą równoodległą). **Kompensacja promienia narzędzia** polega więc na programowaniu ruchu nie po konturze nominalnym, ale jego **ekwidystancie**. Czasami nazywa się to programowaniem po ekwidystancie. Różnica między kompensacją programową i automatyczną polega jedynie na sposobie i miejscu obliczania ekwidystanty. Tworzenie konturu równoodległego może być również wspomagane przez systemy CAD, w których najczęściej wykonuje się dokumentację konstrukcyjną czy technologiczną. Wiele z nich posiada bowiem funkcję tworzenia krzywej odsuniętej (ang. *Offset*), która jest niczym innym jak właśnie ekwidystantą. Należy jeszcze pamiętać o tym, iż każdy kontur posiada **dwie** ekwidystanty (Rys. 101).



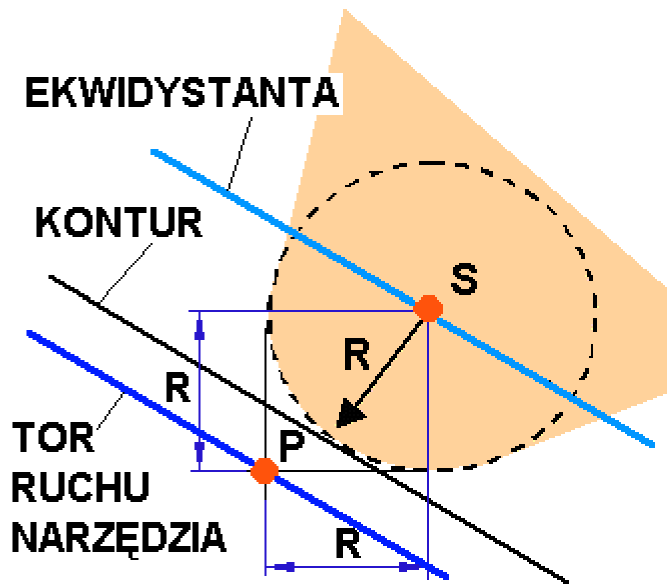
Rys. 100. Zasada kompensowania promienia narzędzia



Rys. 101. Kontur nominalny i jego ekwidystanty

Specyficzną grupę narzędzi stanowią narzędzia tokarskie, dla nich kompensacja promienia jest bardziej złożona. Fragmentem noża tokarskiego, mającego największy wpływ na kształtowanie przedmiotu, jest naroże (krawędź przejściowa pomiędzy głównymi krawędziami skrawającymi) zaokrąglone promieniem *R*. Natomiast punkt kodowy **P**, którego położenie jest programowane, leży zazwyczaj na przecięciu się stycznych do naroża, równoległych do osi układu bazowego (wynika to ze sposobu pomiaru wymiarów narzędzia). Konsekwencją tego jest to, iż punkt kodowy **P** leży **poza** krawędzią skrawającą. Programowanie położenia tak przyjętego punktu

kodowego powoduje w pewnych warunkach powstanie innego konturu po obróbce niż programowany tor ruchu (Rys. 98). Powstająca różnica pomiędzy konturem zadanym (nominalnym) a powstałym po obróbce jest znaczna (zależna od wartości promienia i kąta pochylenia stycznej do konturu w punkcie styku z krawędzią skrawającą) i nie może być pominięta. Aby rozwiązać ten problem przyjmuje się programowanie nie ruch punktu **P**, ale punktu środka okręgu wpisanego w naroże ostrza – punkt kodowy **S** na Rys. 102.



Rys. 102. Kompensacja promienia narzędzia w obróbce tokarskiej

W takiej sytuacji możemy analizować ruch noża tokarskiego jako przemieszczanie się "freza" o środku **S** i promieniu *R* po ekwidystancie. Należy jednak pamiętać, iż w dalszym ciągu punktem programowanym w ruchu narzędzia jest punkt kodowy **P**, co wymaga przesunięcia obliczonej ekwidystanty. Ta zmiana zawsze jest równa wartości promienia *R* narzędzia (Rys. 102), przy czym układ sterowania musi zostać poinformowany o kierunku tego przesunięcia – może ono być dodatnie, ujemne lub równe zero w danej osi, co łącznie daje dziewięć różnych kombinacji położenia obu punktów kodowych. Parametrem, który za to odpowiada, jest kodowe oznaczenie położenia ostrza – Rys. 95. Dla kodów 1-4 przesunięcia odbywa się w obydwu osiach (w kierunku dodatnim lub ujemnym), dla kodów 5-8 tylko wzdłuż jednej osi, a kod 9 jest przypisany narzędziom nie wymagającym kompensacji promienia. Pokazany na Rys. 102 nóż wymaga obu przesunięć w kierunku ujemnym, zatem powinien być scharakteryzowany kodem położenia ostrza równym 3 – por. Rys. 95.

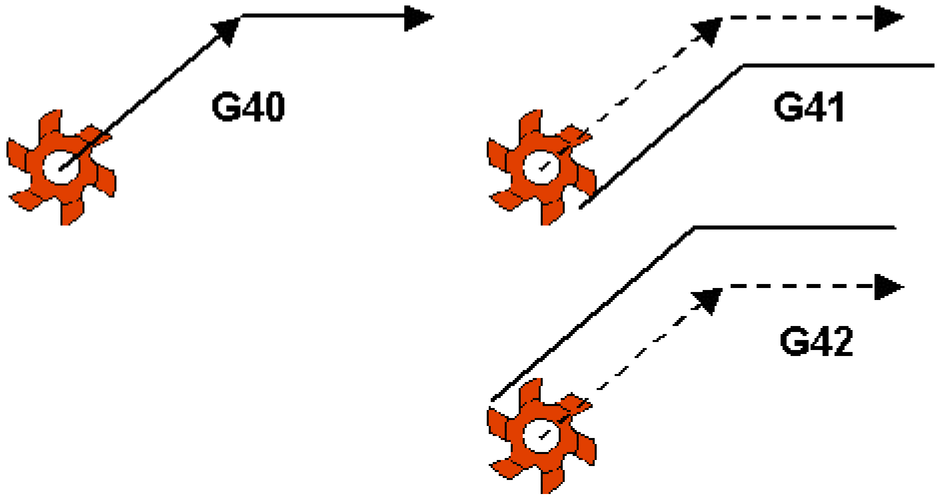
5.3.2. Programowanie automatycznej kompensacji promienia

Automatyczna kompensacja promienia zwalnia zasadniczo programistę z zajmowania się ruchem narzędzia po ekwidystancie. Jego zadaniem jest zapisanie geometrii konturu nominalnego oraz poinformowanie układu sterowania, czy rzeczywisty ruch narzędzia odbywać się ma po zaprogramowanym konturze czy jego ekwidystantach (które układ sterowania musi sam wyznaczyć). Do sterowania sposobem realizacji ruchu zdefiniowano trzy funkcje przygotowawcze (Rys. 103):

G40 – wyłączenie automatycznej kompensacji promienia (ruch punktu kodowego **P** po konturze nominalnym);

G41 – włączenie automatycznej kompensacji promienia po lewej stronie konturu (ruch punktu kodowego **P** lub **S** po ekwidystancie, leżącej po lewej stronie konturu nominalnego patrząc w kierunku ruchu narzędzia);

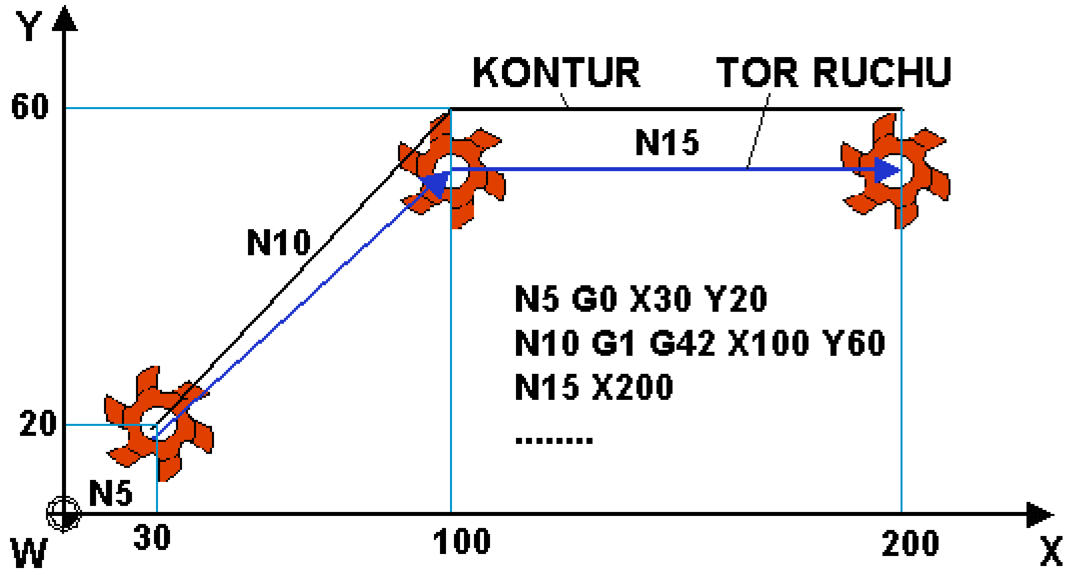
G42 – włączenie automatycznej kompensacji promienia po prawej stronie konturu (ruch punktu kodowego **P** lub **S** po ekwidystancie, leżącej po prawej stronie konturu nominalnego patrząc w kierunku ruchu narzędzia).



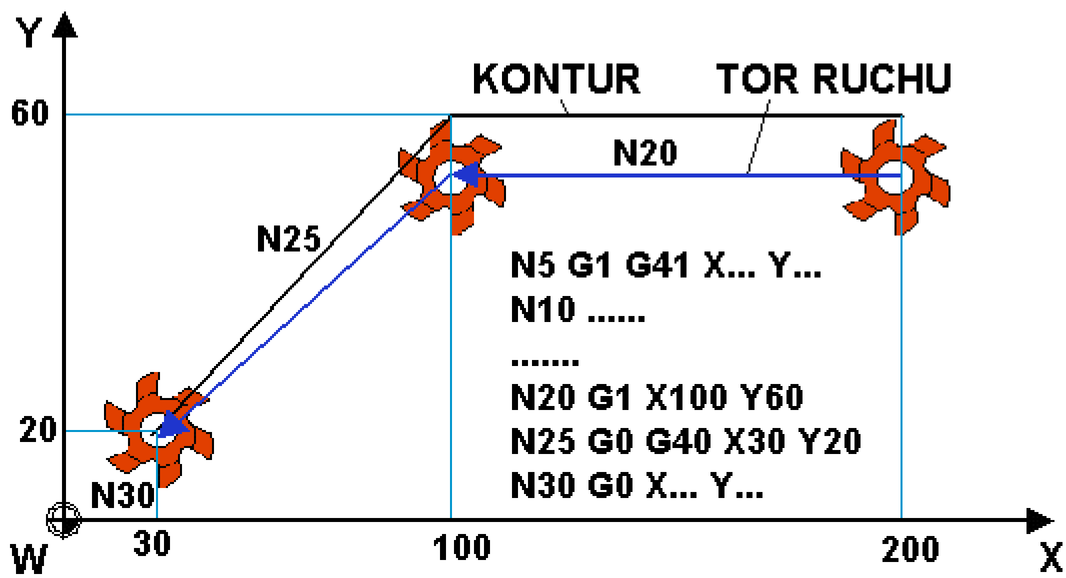
Rys. 103. Programowanie automatycznej kompensacji promienia narzędzia

Ruch w pełni kompensowany jest prosty w programowaniu, newralgicznym momentem jest jednak jego rozpoczęcie i zakończenie. Przejście z ruchu niekompensowanego na kompensowany i odwrotnie nie może się odbyć nagle, zawsze związane to jest z ruchem przejściowym. Ruch ten można opisać następującą zasadą (dla rozpoczęcia kompensacji): narzędzie w bloku, w którym jest włączana automatyczna kompensacja porusza się do punktu na prostej prostopadłej do elementu konturu programowanego w następnym bloku, przechodzącej przez punkt początkowy tego elementu. Czyli ruch w pełni kompensowany występuje dopiero przy następnym fragmencie konturu w stosunku do przejściowego (Rys. 104). Podobną zasadę można sformułować dla przypadku wyłączania kompensacji promienia narzędzia (Rys. 105).

Blok przejściowy musi być programowany z interpolacją liniową (zalecane) lub punktową. Nie może natomiast wystąpić tu interpolacja kołowa lub inna. Podobna sytuacja następuje jeżeli następuje zmiana promienia narzędzia przez zaprogramowanie innego rejestru narzędziowego lub zmiana kierunku kompensacji (G41 na G42 i odwrotnie). Ogólnie przy programowaniu ruchu z automatyczną kompensacją promienia należy stosować szczególną ostrożność, gdyż nie zawsze jesteśmy w stanie dokładnie przewidzieć rzeczywistego toru ruchu narzędzia, który czasem może być dość niespodziewany. Wynika to z dużej złożoności algorytmów obliczania ekwidystanty, których twórcy nie zawsze przewidzieli wszystkie możliwe sytuacje.



Rys. 104. Programowanie rozpoczęcia automatycznej kompensacji promienia narzędzia



Rys. 105. Programowanie zakończenia automatycznej kompensacji promienia narzędzia

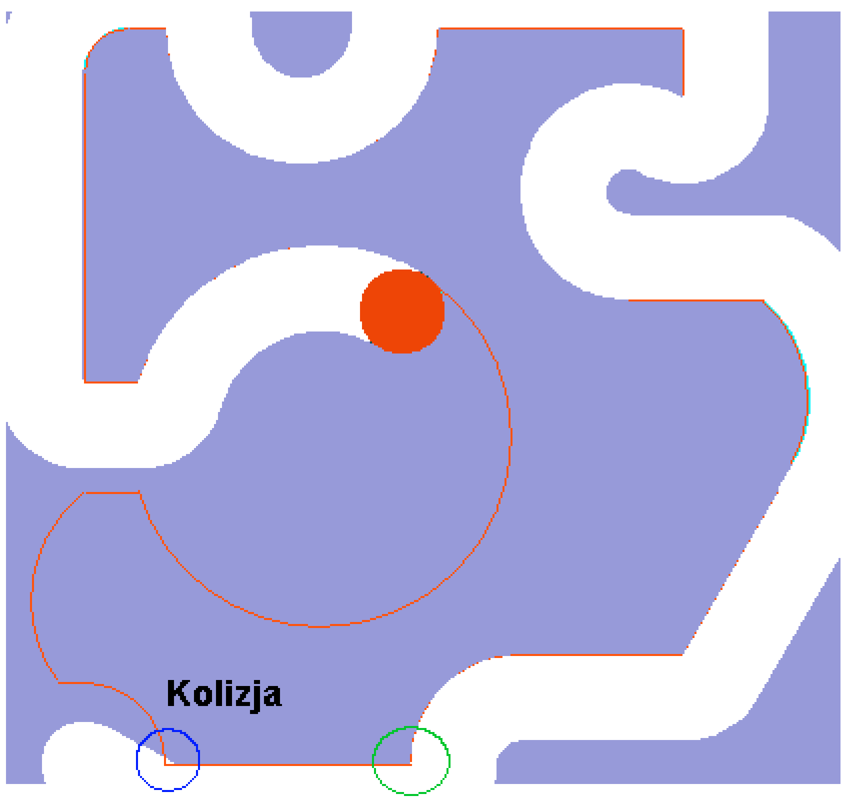
5.3.3. Przykład

Dla przykładu frezowania konturu (rozdz. 3.4.1) uzupełnić program sterujący o automatyczną kompensację promienia.

Rozwiązanie:

```
%_N_EX06_MPF
; 22-08-2003
N5 G54 G71 G90 G94 G17
N10 S800 F100 T1 D1 M3 M8 M6
N15 G0 X0 Y0
N20 Z3
N25 G1 Z-5
N30 G42 X15              ;START KOMPENSACJI
```

```
N35 X60
N40 G2 I20 J0 AR=90
N45 G1 X110
N50 X130 Y54.641
N55 CT X125 Y85
N60 G91 G1 X-25
N65 G2 J20 AR=210
N70 G1 G90 Y135
N75 X=IC(-45)
N80 G2 X15 AR=180
N85 G1 X0 RND=8
N90 G1 Y70
N95 X10
N100 CIP Y=IC(-20) I1=79 J1=60
N105 G1 X0
N110 G3 X-5 Y15 I=AC(15) J-20
N115 G1 X0
N120 G2 J-15 AR=90
N125 G1 G40 Y-20       ;KONIEC KOMPENSACJI
N130 G0 Z100
N135 G53 T0 D0 G0 X300 Y300 Z200 M9 M5 M30
```



Rys. 106. Symulacja programu sterującego z wykorzystaniem automatycznej kompensacji promienia

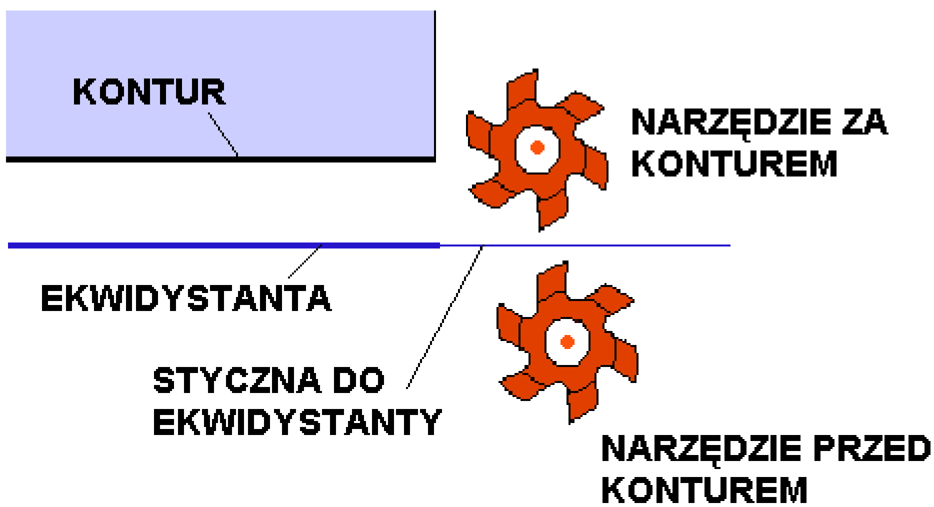
5.3.4. Inne funkcje sterujące automatyczną kompensacją promienia

Jeżeli dobrze przyjrzeć się wynikowi symulacji programu z przykładu z poprzedniego rozdziału to można zauważyć naruszenie konturu przedmiotu obrabianego (w niebieskim kółku na Rys. 106). Pozornie wszystko zostało zaprogramowane w sposób prawidłowy, a mimo tego program jest błędny. Błąd ten jest konsekwencją ustawień dotyczących automatycznej kompensacji, które zostaną omówione w niniejszym rozdziale i dzięki którym można go wyeliminować.

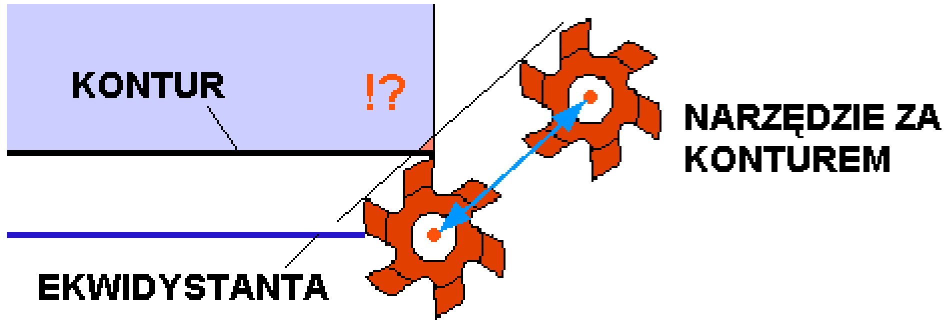
Przy rozpoczynaniu czy kończeniu ruchu kompensowanego rozróżnia się dwa przypadki pozycjonowania narzędzia (Rys. 107):

1. Narzędzie przed konturem – narzędzie i kontur po przeciwnych stronach linii granicznej (bierze się pod uwagę położenie punktu kodowego narzędzia);
2. Narzędzie za konturem – narzędzie i kontur po tej samej stronie linii granicznej.

Linią graniczną jest tu styczna do ekwidystanty w punkcie początkowym/końcowym kompensacji i sama ekwidystanta. Jeżeli rozpoczęcie lub zakończenie ruchu kompensowanego odbywa się dla narzędzia przed konturem to nie ma negatywnych skutków takiego ruchu. Inaczej jest w przypadku narzędzia za konturem. Wtedy może dojść do ruchu kolizyjnego (Rys. 108), jak to miało miejsce w omawianym przykładzie.



Rys. 107. Położenie narzędzia przy rozpoczęciu/zakończeniu ruchu kompensowanego



Rys. 108. Ruch kolizyjny narzędzia przy rozpoczęciu/zakończeniu kompensacji promienia narzędzia – narzędzie za konturem

W układzie sterowania Sinumerik 840D nadzór nad sposobem rozpoczynania/kończenia ruchu kompensowanego pełnią następujące funkcje przygotowawcze (modalne):

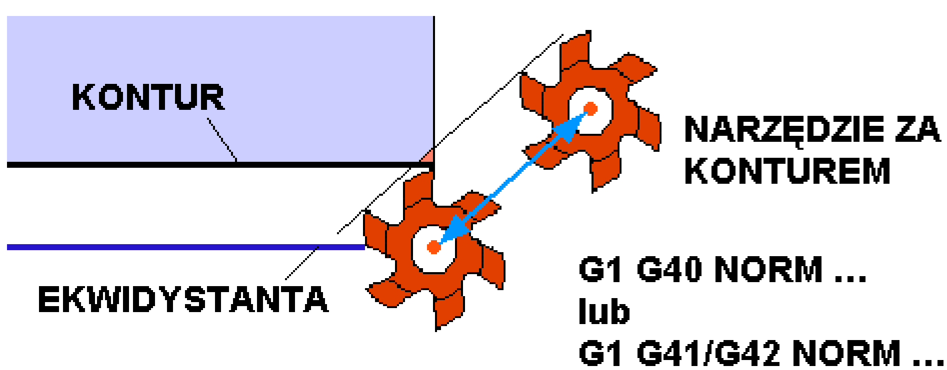
NORM – (ang. *NORMal*) narzędzie nie wykonuje żadnych dodatkowych ruchów i po najkrótszym torze przemieszcza się od końca ekwidystanty do kolejnego punktu programowanego (przy zakończeniu kompensacji) lub od aktualnego położenia do początku ekwidystanty (przy rozpoczęciu kompensacji) (Rys. 109); **funkcja ta na ogół jest przyjmowana jako domyślna**;

KONT – (niem. *KONTurpunkt*) narzędzie wykonuje obejście punktu początkowego/końcowego ekwidystanty jeżeli znajduje się za konturem (Rys. 110); jeżeli znajduje się przed konturem to jego zachowanie jest analogiczne jak dla funkcji NORM.

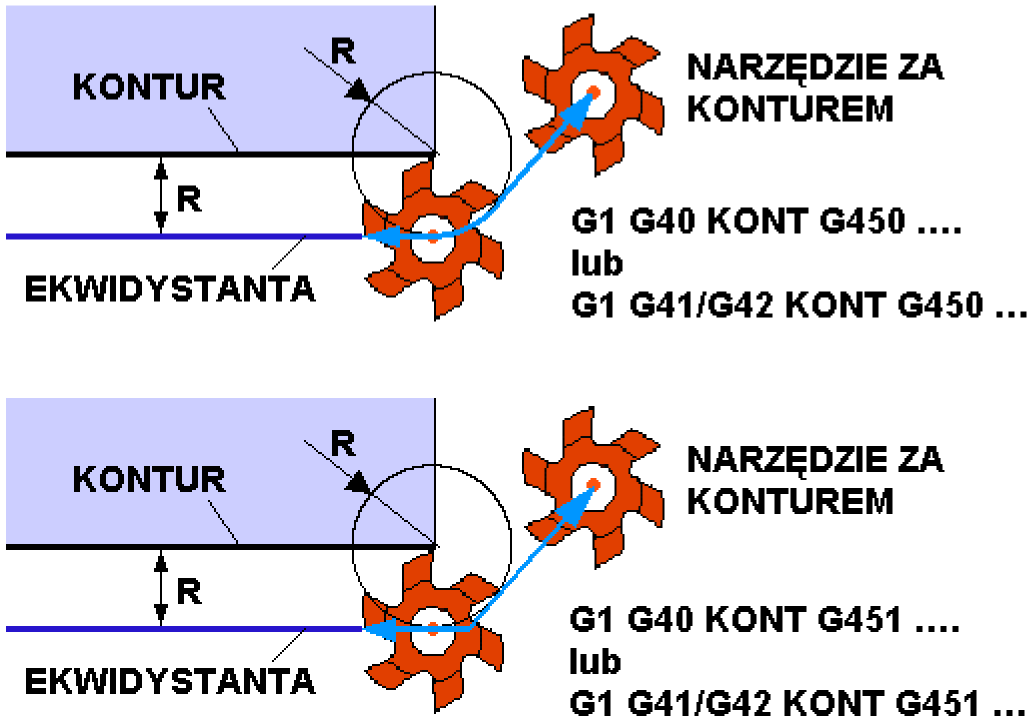
Sposób obejścia jest regulowany przez dwie kolejne funkcje przygotowawcze (modalne):

G450 – obejście po łuku o środku w punkcie początkowym/końcowym programowanego konturu i promieniu narzędzia (Rys. 110);

G451 – obejście przez punkt przecięcia stycznych do okręgu o środku w punkcie początkowym/końcowym programowanego konturu i promieniu narzędzia (Rys. 110).



Rys. 109. Programowanie rozpoczęcia/zakończenia kompensacji promienia narzędzia z użyciem funkcji **NORM**

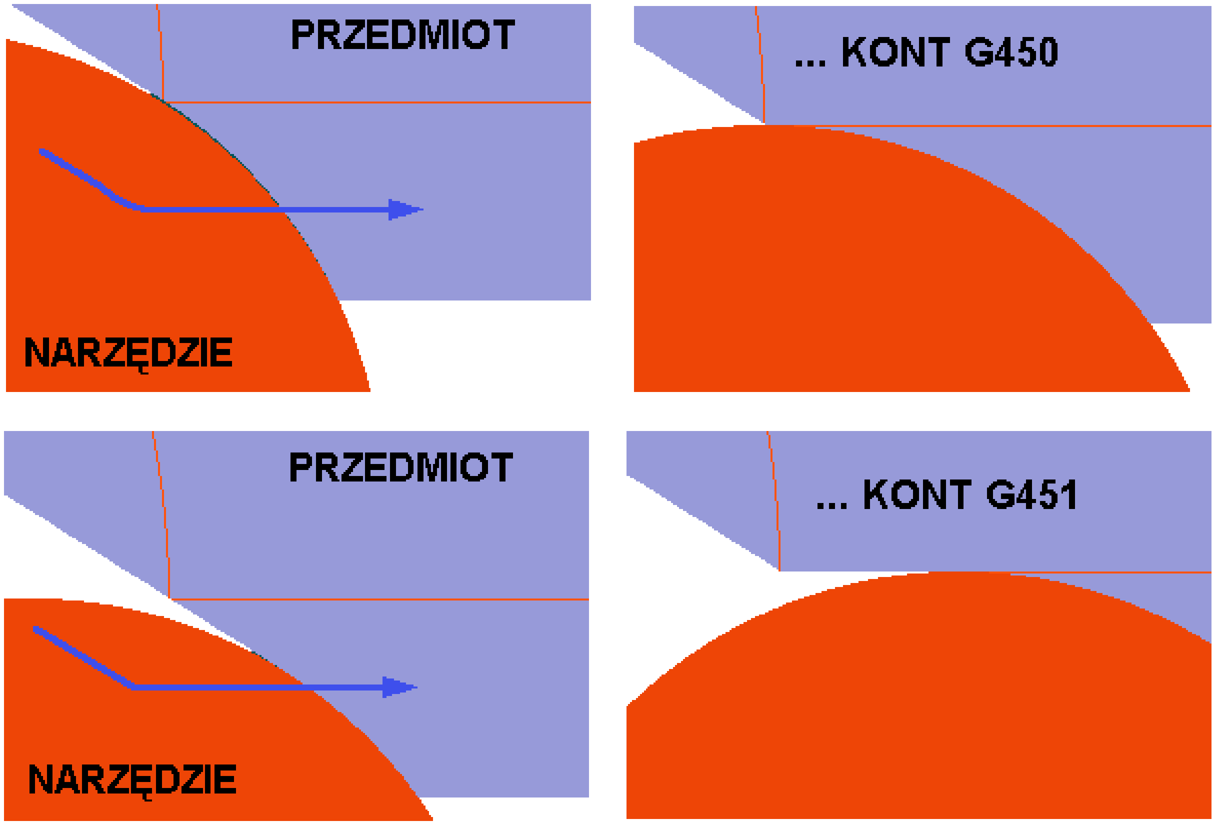


Rys. 110. Programowanie rozpoczęcia/zakończenia kompensacji promienia narzędzia z użyciem funkcji **KONT** + **G450/G451**

Przykład:
Program przedstawiony w rozdz. 2.3.3 uzupełnić o funkcje kontroli obejścia punktu początkowego/końcowego kompensacji w celu wyeliminowania sytuacji kolizyjnej (naroże oznaczone niebieskim kółkiem na Rys. 106).
Rozwiązanie:
Dokonano zmiany bloku N05, przyjmując dwa warianty:

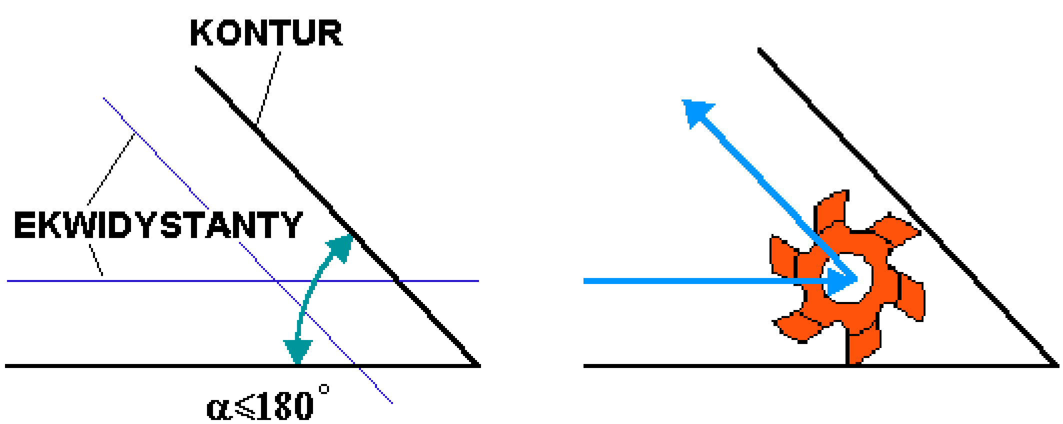
1. N5 G54 G71 G90 G94 G17 KONT G450
2. N5 G54 G71 G90 G94 G17 KONT G451

W wyniku symulacji zmodyfikowanych programów uzyskano następujący efekt przy rozpoczęciu kompensacji, przedstawiony na Rys. 111. Jak łatwo zauważyć, przy zastosowaniu funkcji G450 uzyskuje się ciągły kontakt narzędzia z obrabianym materiałem, przy funkcji G451 wprost przeciwnie – narzędzie przy obejściu naroża traci kontakt z materiałem. Wybór jednej z obu funkcji powienien zatem być uwarunkowany wymaganiami technologicznymi i doświadczeniem programisty.



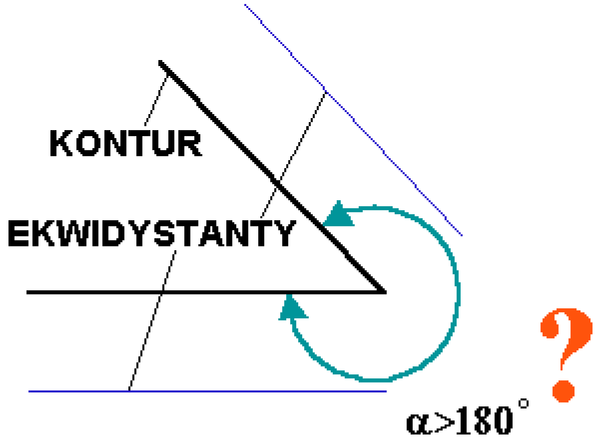
Rys. 111. Wyniki symulacji zmodyfikowanego programu z rozdz. 2.3.3

Funkcje G450/G451 sterują również sposobem generowania ekwidystanty przy przejściach pomiędzy kolejnymi fragmentami konturu. Są tu dwa przypadki: jeżeli kąt pomiędzy stycznymi do sąsiednich fragmentów konturu jest mniejszy lub równy 180° (tzw. naroże wewnętrzne) oraz jeżeli ten kąt jest większy niż 180° (tzw. naroże zewnętrzne). W pierwszym przypadku wyznaczany jest punkt przecięcia ekwidystant, stanowiący punkt zwrotny ruchu (Rys. 112).



Rys. 112. Tor ruchu po ekwidystancie naroża wewnętrznego

Bardziej skomplikowany jest przypadek drugi. Wtedy pomiędzy ekwidystantami do kolejnych fragmentów konturu istnieje przerwa (niezdefiniowany odcinek toru narzędzia) – Rys. 113.

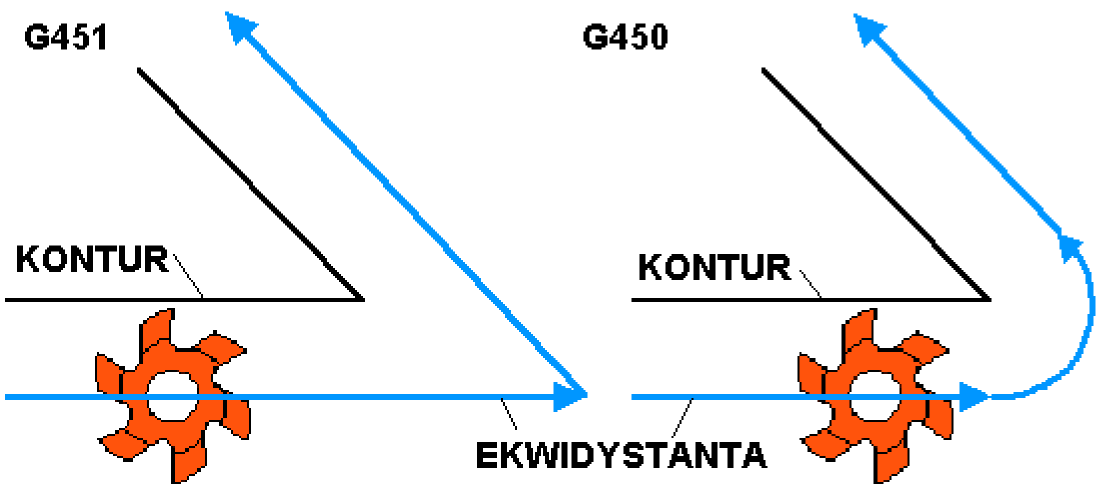


Rys. 113. Niezdefiniowany tor ruchu po ekwidystancie naroża zewnętrznego

Wypełnienie tego fragmentu w układzie Sinumerik 840D jest zależne od wspomnianych już funkcji przygotowawczych G450/G451. Działają one w podobny sposób, jak przy rozpoczęciu/zakończeniu kompensacji (Rys. 114):

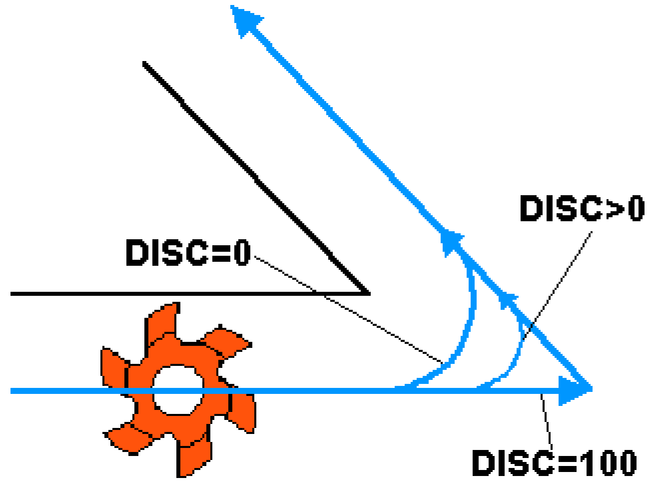
G450 – tor ruchu narzędzia pomiędzy ekwidystantami przebiega po łuku o środku w punkcie przecięcia się obu fragmentów konturu;

G451 – tor ruchu narzędzia przebiega po przedłużeniu ekwidystant aż do ich punktu przecięcia.



Rys. 114. Programowanie toru ruchu po ekwidystancie naroża zewnętrznego

Istnieje funkcja pozwalająca na zaprogramowanie pośredniej postaci toru ruchu przy obejściu naroży konturu – **DISC** (ang. *DIStort transition Circle*). Jest to funkcja modalna, działająca przy aktywnej funkcji G450. Jej wartość należy do przedziału od 0 do 100. W zależności od wartości funkcji DISC można uzyskać obejście po łuku, do przecięcia ekwidystant lub pośrednie – Rys. 115.



Rys. 115. Kontrola toru narzędzia przy kompensacji za pomocą funkcji DISC

Przy rozpoczynaniu i kończeniu ruchu kompensowanego można również wykorzystać tzw. miękkie dosunięcie/odsunięcie narzędzia, polegające na wykonaniu dodatkowych ruchów, zapewniających płynne rozpoczęcie lub zakończenie kontaktu narzędzia z obrabianym materiałem. W układzie sterowania Sinumerik 840D przewidziano trzy typy takich ruchów, programowanych przez sześć funkcji przygotowawczych:

G147 – dosunięcie po prostej, przedłużenie stycznej do ekwidystanty o wartość programowaną pod adresem **DISR** (ang. *DIStort Radius*) – Rys. 116;

G247 – dosunięcie po ¼ okręgu o promieniu programowanym pod adresem **DISR** – Rys. 117;

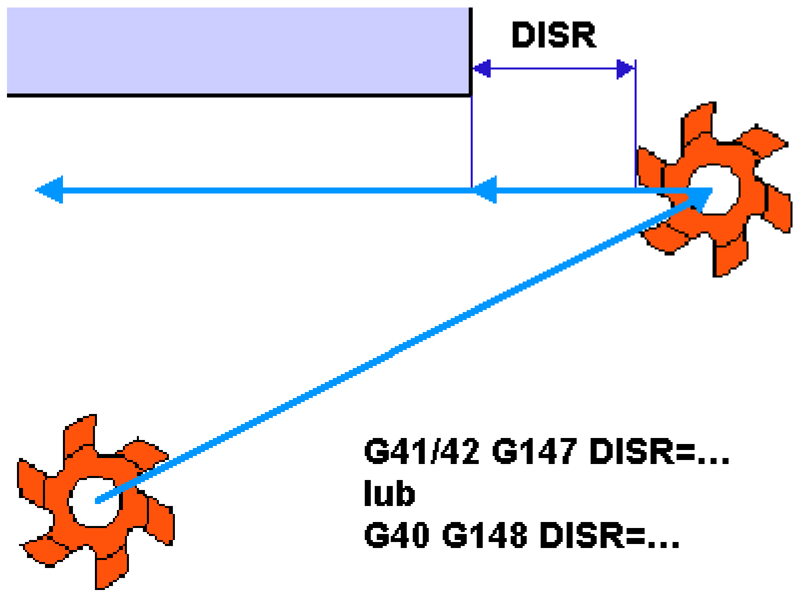
G347 – dosunięciepo ½ okręgu o promieniu programowanym pod adresem **DISR** – Rys. 118;

G148 – odsunięcie po prostej, przedłużenie stycznej do ekwidystanty o wartość programowaną pod adresem **DISR** – Rys. 116;

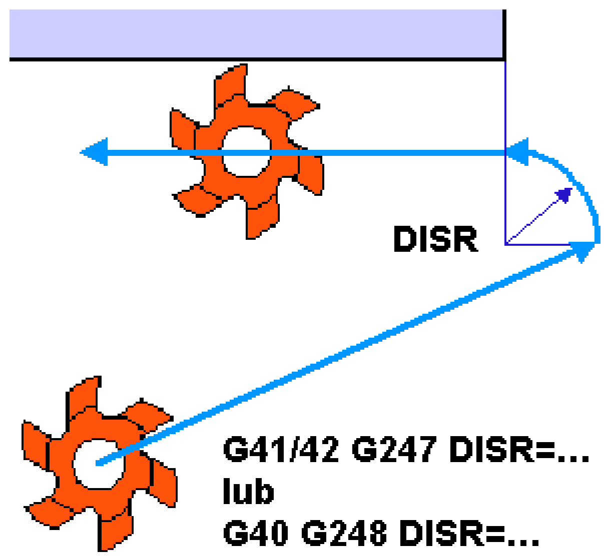
G248 – odsunięcie po ¼ okręgu o promieniu programowanym pod adresem **DISR** – Rys. 117;

G348 – odsunięcie po ½ okręgu o promieniu programowanym pod adresem **DISR** – Rys. 118.

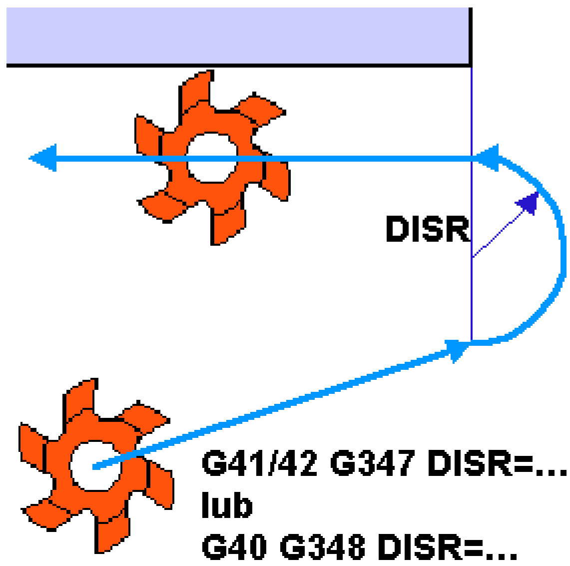
Programowanie miękkiego dosunięcia/odsunięcia stosuje się np. ze względu na obróbkę powierzchni wewnętrznych. Wtedy brak jest dobiegu lub wybiegu narzędzia poza materiałem (jak przy obróbce powierzchni zewnętrznych). Dosuwanie lub odsuwanie narzędzia musi odbywać się wzdłuż krawędzi konturu, co może powodować powstanie śladów na wskutek odkształceń narzędzia. Żeby tego uniknąć dosuwa się lub odsuwa narzędzie stopniowo zagłębiając go w materiał i tym samym łagodząc skutki jego odkształceń.



Rys. 116. Programowanie miękkiego dosunięcia/odsunięcia po linii prostej (G147/G148)



Rys. 117. Programowanie miękkiego dosunięcia/odsunięcia po ćwierćokręgu (G247/G248)



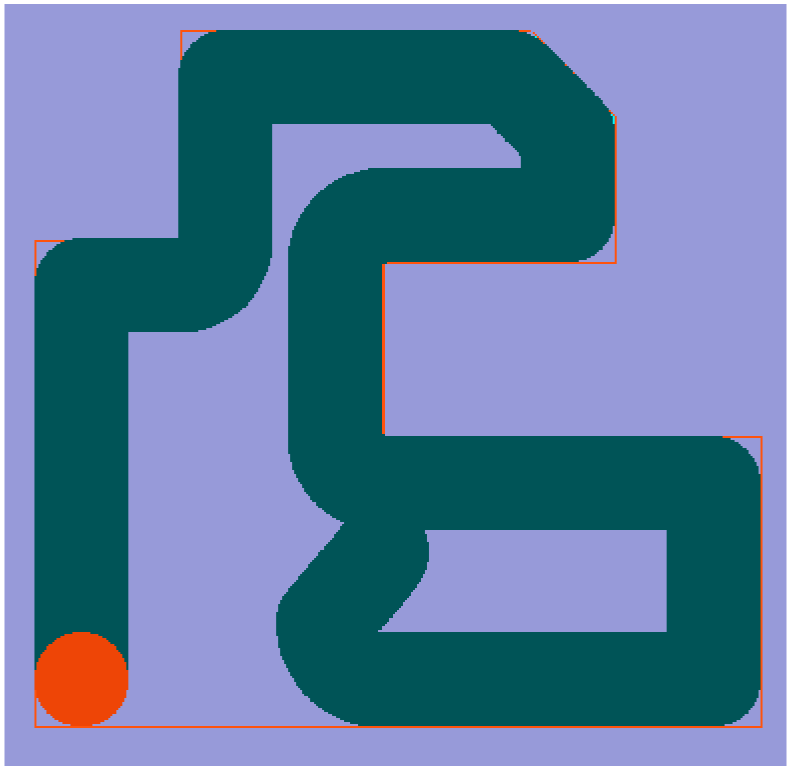
Rys. 118. Programowanie miękkiego dosunięcia/odsunięcia po półokręgu (G347/G348)

5.3.5. Przykład

Dla frezowania konturu z rozdz. 3.3.1 uwzględnić automatyczną kompensację promienia narzędzia, zakładając obróbkę wewnętrzną. Wykorzystać miękkie dosunięcie i odsunięcie narzędzia po ćwierćokręgu. Naroża zewnętrzne wykonać z obejściem po promieniu.

Rozwiązanie:

```
%_N_EX07_MPF
; 23-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S1500 F250 M3 M8 M6
N15 G0 X60 Y30 ; NAJAZD NAD WNETRZE KONTURU
N20 Z3
N25 G1 Z-5  ; WEJSCIE W MATERIAL
N30 G41 G247 Y0 DISR=10 ; START KOMPENSACJI
N35 X125
N40 Y50
N45 X60
N50 Y80
N55 X100
N60 Y105.4
N65 X85.4 Y120
N70 X25
N75 Y84
N80 X0
N85 Y0
N90 X60
N95 G40 G248 Y30 DISR=10 ; KONIEC KOMPENSACJI
N100 G0 Z100
N105 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N110 M30
```



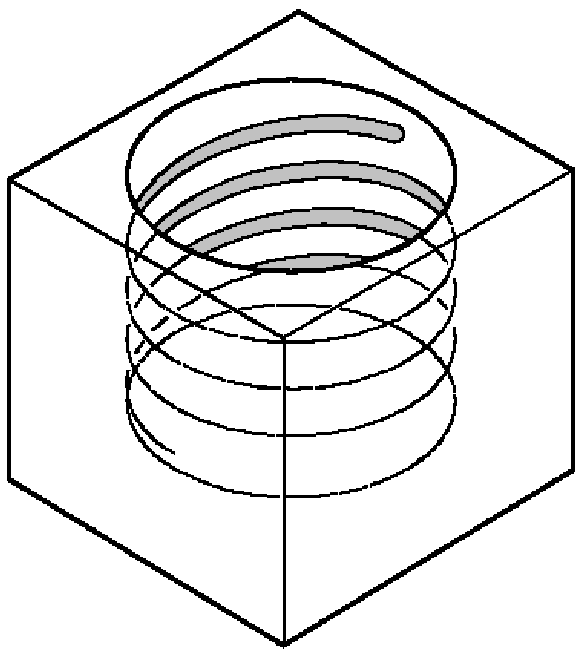
Rys. 119. Symulacja programu sterującego z uwzględnieniem automatycznej kompensacji promienia oraz miękkiego dosunięcia/odsunięcia narzędzia

6. OBRÓBKA GWINTÓW NA OBRABIARKACH CNC

Gwint (oraz inne powierzchnie bazujące na zarysie spiralnym) to specyficzny rodzaj powierzchni obrabianych, często spotykany w przedmiotach obrabianych. Różne są techniki obróbki gwintów – toczenie, frezowanie, nacinanie narzędziami kształtowymi – a tym samym różne funkcje dla nich muszą być stosowane. W niniejszym rozdziale obszernie przedstawiono ich działanie i zastosowanie.

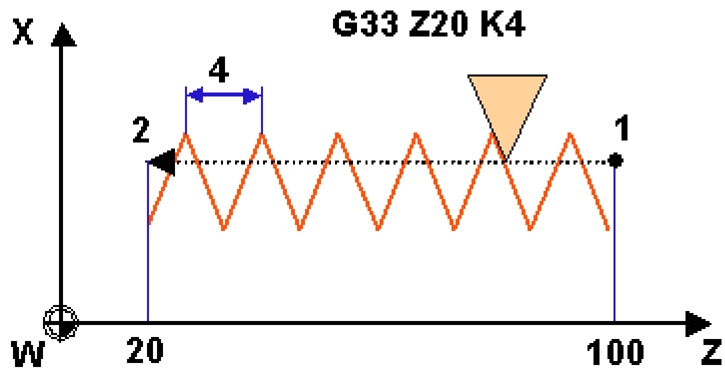
6.1. Interpolacja spiralna o stałym skoku G33

Obróbka powierzchni spiralnych wiąże się przede wszystkim z nacinaniem gwintu (Rys. 120). Działanie funkcji G33 (należy do grupy funkcji G0, G1, ...) polega na stworzeniu "elektronicznej gitary", sprzęgającej ruch obrotowy wrzeciona z posuwami liniowymi, dając w efekcie ruch po krzywej spiralnej (helikoidalnej). Pozwala to na toczenie gwintów czy zarysów spiralnych przy wielokrotnych przejściach narzędzia kształtowego (zarys narzędzia musi być dopasowany do zarysu gwintu).

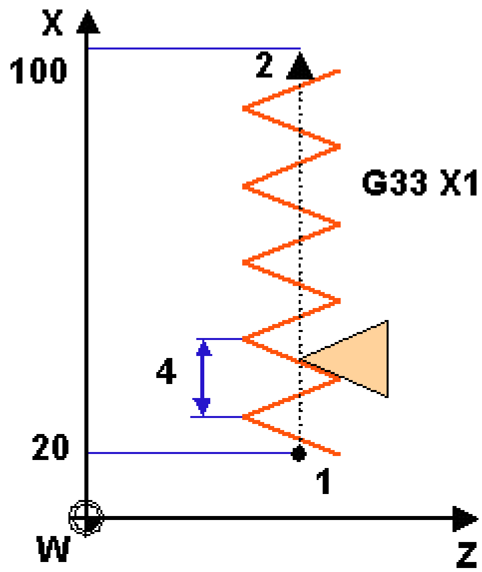


Rys. 120. Gwint jako przykład wykorzystania interpolacji spiralnej

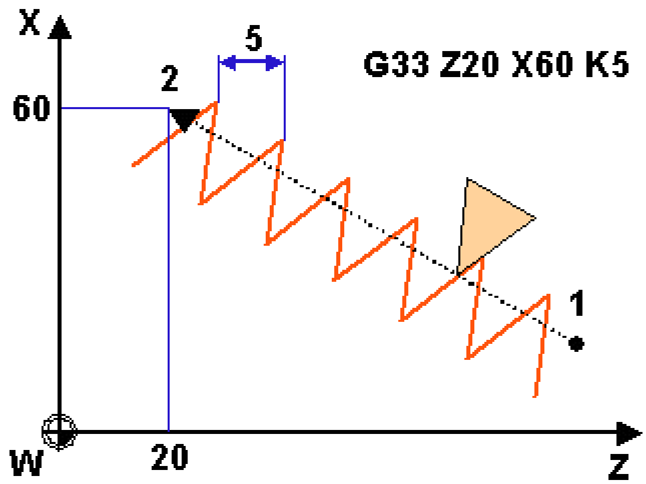
Tak jak w innych rodzajach interpolacji musi być podany punkt końcowy ruchu przy zachowaniu wszystkich obowiązujących zasad. Dodatkową informacją jest skok spirali (zawsze jako liczba dodatnia), programowany za pomocą parametrów interpolacji I, J, K odpowiednio do osi, wzdłuż której odbywa się ruch (Rys. 121 i Rys. 122). Jeżeli interpolacja spiralna obejmuje ruch w dwóch osiach liniowych to podaje się tylko skok spirali tylko wzdłuż jednej osi (w innych układach sterowania często jest inaczej), przy czym jest to ta oś, względem której tor ruchu tworzy mniejszy kąt (Rys. 123 i Rys. 124).



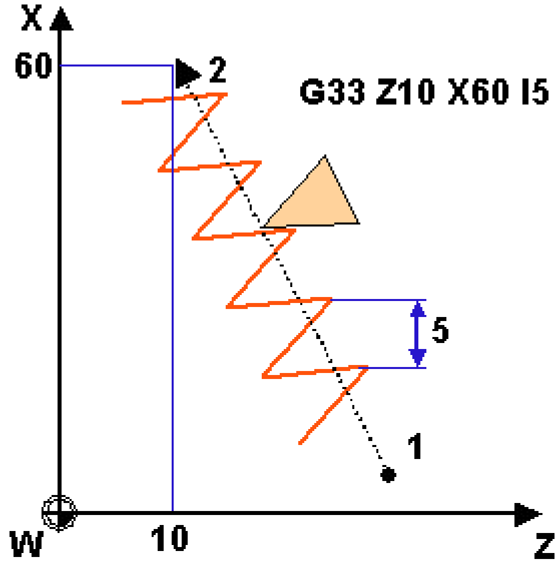
Rys. 121. Programowanie interpolacji spiralnej wzdłuż osi Z



Rys. 122. Programowanie interpolacji spiralnej wzdłuż osi X

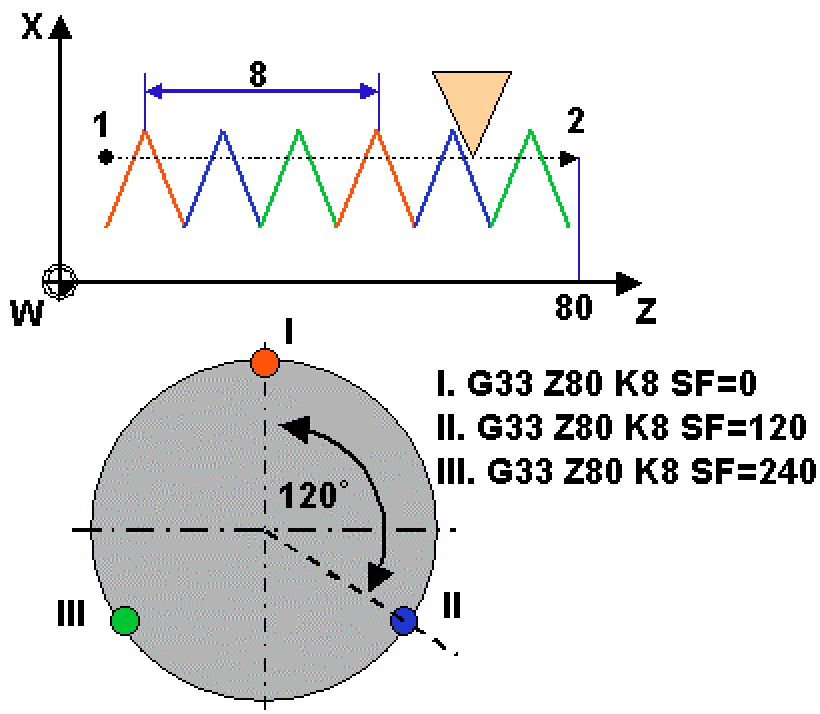


Rys. 123. Programowanie interpolacji spiralnej pod kątem (parametr interpolacji w osi Z)



Rys. 124. Programowanie interpolacji spiralnej pod kątem (parametr interpolacji w osi X)

Przy nacinaniu gwintów wielozwojnych istnieje konieczność zmiany kątowego położenia wrzeciona przy rozpoczęciu ruchu. Domyślnie odbywa się to przy położeniu kątowym wrzeciona pobieranym z danych maszynowych. Przy pomocy adresu **SF** (ang. *Starting point oFfset*) można zaprogramować inne położenie kątowe wrzeciona. Rys. 125 przedstawiono przykład zaprogramowania fragmentu obróbki gwintu 3-zwojnego przy wykorzystaniu adresu SF.

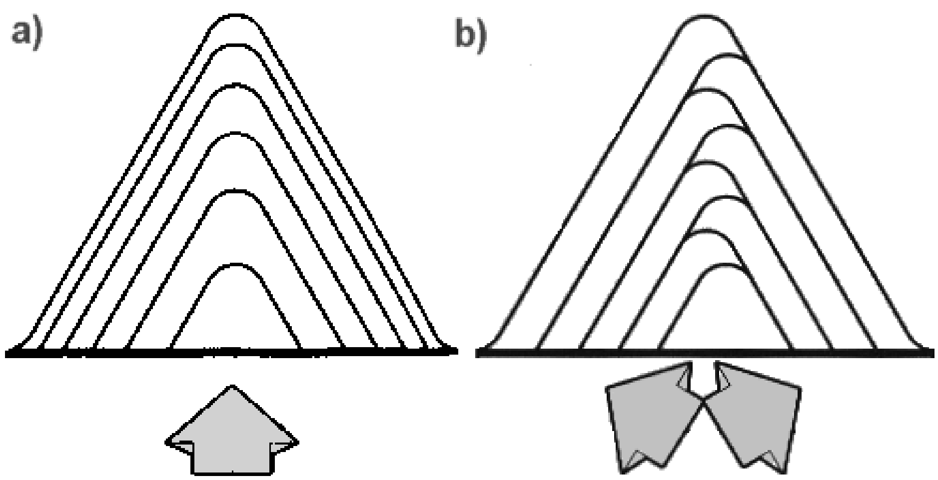


Rys. 125. Programowanie obróbki gwintu wielozwojnego

Przy obróbce gwintu nożem tokarskim należy zwykle wykonać kilka przejść (w tym również przejść wykańczających). Pojawia się zatem problem liczby

niezbędnych przejść i podziału naddatku obróbkowego. Liczba przejść jest uzależniona przede wszystkim od podziałki gwintu oraz jego rodzaju (zewnętrzny, wewnętrzny, metryczny, calowy itp.). Wytyczne dotyczące liczby przejść można znaleźć np. w katalogach narzędziowych, gdzie znajdują się również inne informacje o obróbce gwintów.

Podział naddatku obróbkowego musi uwzględniać rodzaj materiału i powodować jak najbardziej korzystne warunki spływu wióra. W praktyce spotkać można kilka typowych sposobów podziału naddatku przy obróbce gwintu – Rys. 126, Rys. 127. Metoda z Rys. 126a wymaga jednoczesnej pracy dwóch krawędzi, co prowadzi do znacznych naprężeń w obszarze wierzchołka noża, jednak powoduje równomierne zużycie krawędzi skrawających. Ten typ obróbki zalecany jest zatem w przypadku obróbki materiałów kruchych, dających krótki, łamliwy wiór (np. brąz, mosiądz), dla materiałów przejawiających samoutwardzanie (np. stale nierdzewne) lub gwintów o małych podziałkach. Pozostałe metody, a szczególnie wcinanie boczne (Rys. 127), ma za zadanie prowadzić obróbkę możliwie jedną krawędzią, przy czym metoda naprzemienna daje bardziej równomierne zużycie krawędzi (zalecana jest przede wszystkim dla gwintów o dużych podziałkach). Ten typ wcinania jest zalecany dla materiałów ciągliwych, o wiórze spiralnym, śrubowym czy pasmowym (np. stale konstrukcyjne).

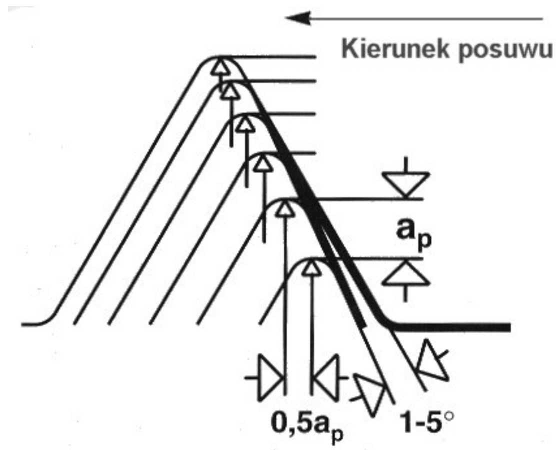


Rys. 126. Podział naddatku obróbkowego przy gwintowaniu: a) wgłębny promieniowy, b) wgłębny naprzemienny [Sandvik Coromant]

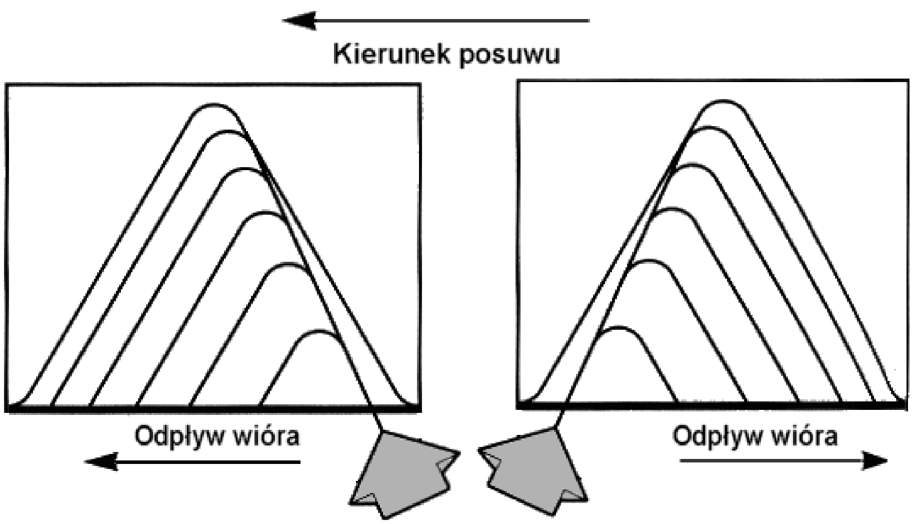
Przy bocznym podziale naddatku można wcinać się dokładnie wzdłuż powierzchni bocznej gwintu. Jednak z uwagi na dokładność pozycjonowania narzędzia, jak i nieuniknione przy znacznych siłach obróbki jego odkształcenia, na powierzchni bocznej gwintu mogą być widoczne ślady obróbki, co znacznie pogarsza jakość gwintu. Zaleca się zatem wcinanie z zachowaniem pewnego niewielkiego kąta w stosunku do powierzchni bocznej gwintu (2÷5°), umożliwiające uzyskanie dobrej jakości powierzchni gwintu. Przy tym sposobie obróbki należy również pamiętać o tym, że nóż powinien „pchać” wiór do przodu (Rys. 128).

Po ustaleniu liczby przejść i kierunku wcinanie przy kolejnych przejściach należy określić kolejne głębokości obróbki. Stosuje się tu dwie strategie (Rys. 129): stały dosuw (narzędzie wcina się o stałą wartość w kierunku promieniowym) oraz stały

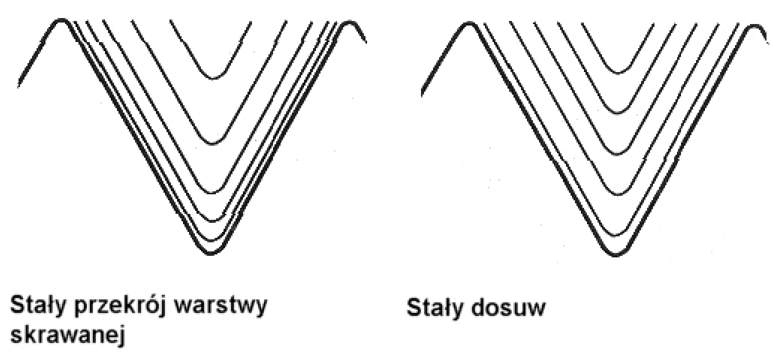
przekrój warstwy skrawanej (narzędzie wcina się o zmienną, malejącą wartość w kierunku promieniowym). Stały dosuw może być stosowany dla gwintów o małych podziałkach, dla pozostałych należy stosować bardziej korzystny stały przekrój warstwy skrawanej.



Rys. 127. Boczny podział naddatku obróbkowego przy gwintowaniu [Sandvik Coromant]

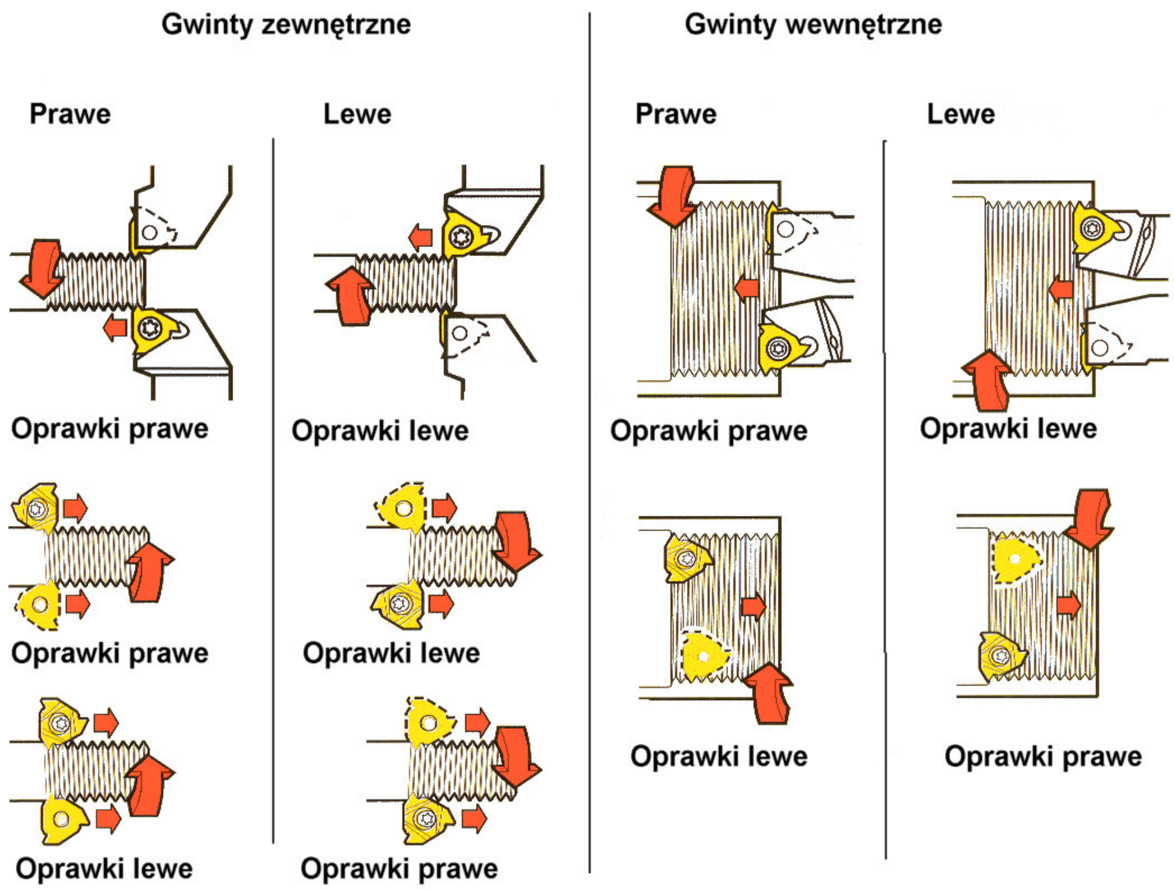


Rys. 128. Kierunek wcinania przy bocznym podziale naddatku obróbkowego [Sandvik Coromant]



Rys. 129. Podział naddatku przy gwintowaniu z uwagi na kolejne głębokości obróbki [Sandvik Coromant]

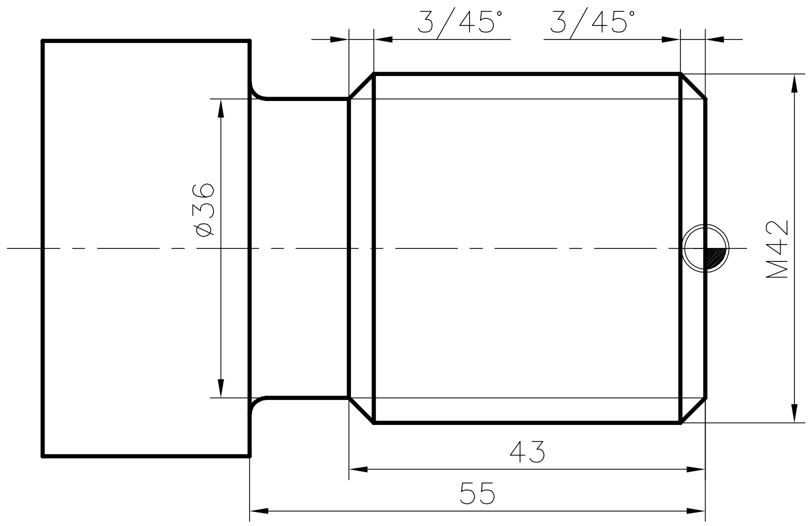
W nacinaniu gwintów na tokarce szczególnie ważne jest odpowiednie skojarzenie kierunku obrotów wrzeciona i kierunku posuwu, ponieważ tylko wtedy uzyskamy gwint prawo- lub lewozwojny. W sposób schematyczny to skojarzenie przedstawiono na Rys. 130. Ze względu na złożoność obliczeniową i znaczną liczbę bloków wymaganych do zaprogramowania nacinania gwintu w praktyce stosuje się cykle nacinania gwintów, o czym traktuje rozdz. 7.



Rys. 130. Schemat doboru parametrów obróbki (kierunek obrotów wrzeciona i kierunek posuwu) przy nacinaniu gwintów na tokarce [Sandvik Coromant]

6.1.1. Przykład

Zaprogramować obróbkę gwintu M42 (Rys. 131). Zastosować n = 14 przejść zgrubnych i jedno tzw. puste, z zerowym naddatkiem. Przyjąć obróbkę z wcinaniem pod kątem φ = 28° względem prostopadłej do osi gwintu. Głębokości kolejnych przejść obliczyć korzystając z warunku jednakowych przekrojów warstwy skrawanej.



Rys. 131. Szkic przedmiotu do przykładu obróbki gwintu toczeniem

Rozwiązanie:

Gwint M42 posiada skok P = 4.5, średnica wewnętrzna d_3 = 36.081. Dla obliczenia głębokości kolejnych przejść i przesunięć w osi Z posłużono się następującymi zależnościami (Rys. 132):

$$h_i = h \cdot \sqrt{\frac{i}{n}}; \quad i = 1,...,n \tag{9}$$

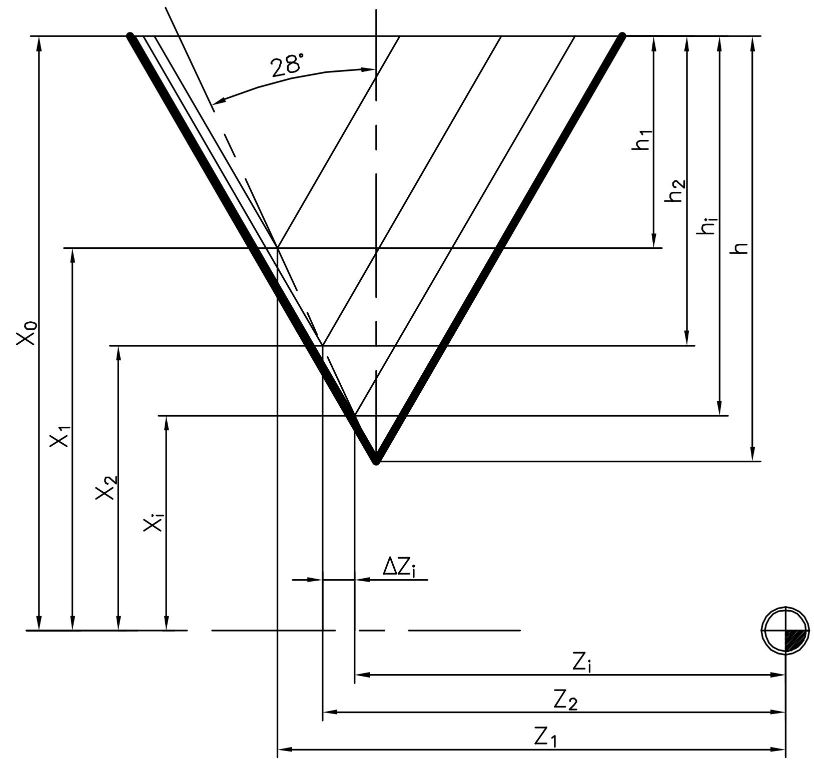
Powyższa zależność w uproszczony sposób pozwala na zachowanie stałego, równomiernego przekroju warstwy skrawanej.

$$X_i = X_0 - h_i; \quad i = 1,...,n \tag{10}$$

$$\Delta Z_i = (X_{i-1} - X_i) \cdot \mathrm{tg}(\varphi); \quad i = 2,...,n \tag{11}$$

$$Z_i = Z_{i-1} + \Delta Z_i; \quad i = 2,...,n \tag{12}$$

Dla uproszczenia przyjęto wymiarowanie w osi X na promieniu. Wyniki obliczeń, wymaganych przy programowaniu, najlepiej zestawić w formie tabeli (Tabl. 2). Dla obróbki gwintu prawozwojnego zgodnie z Rys. 130 przyjęto kierunek posuwu od wrzeciona do konika. Na Rys. 133 przedstawiono fragment symulacji obróbki.



Rys. 132. Wyznaczanie współrzędnych punktów startu dla kolejnych przejść

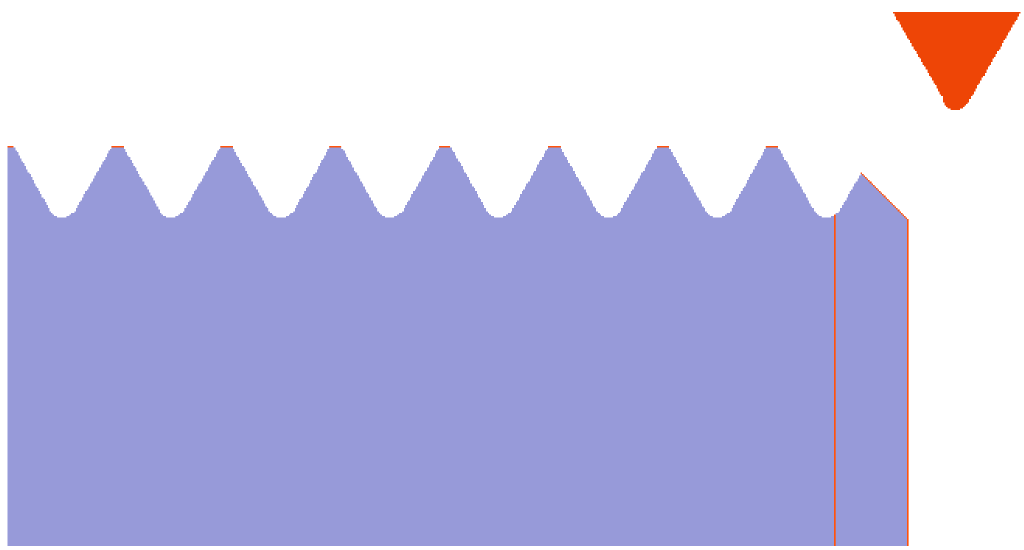
Tabl. 2. Wyniki obliczeń dla kolejnych przejść przy toczeniu gwintu (przyjęte wielkości zadane $X_0 = 21$, $Z_1 = -45$, $h = 2.960$, $\varphi = 28°$, $n = 14$)

i	h_i	X_i	ΔZ_i	Z_i
1	0.7911	20.2089		-45.0000
2	1.1188	19.8812	0.1742	-44.8258
3	1.3702	19.6298	0.1337	-44.6921
4	1.5822	19.4178	0.1127	-44.5794
5	1.7689	19.2311	0.0993	-44.4801
6	1.9378	19.0622	0.0898	-44.3903
7	2.0930	18.9070	0.0826	-44.3077
8	2.2375	18.7625	0.0768	-44.2309
9	2.3733	18.6267	0.0722	-44.1587
10	2.5017	18.4983	0.0683	-44.0905
11	2.6238	18.3762	0.0649	-44.0256
12	2.7404	18.2596	0.0620	-43.9635
13	2.8523	18.1477	0.0595	-43.9040
14	2.9600	18.0400	0.0573	-43.8468

Program sterujący:

```
%_N_EX11_MPF
; 30-08-2003
N5 G40 G54 G71 G90 G96 DIAMON KONT G450
N10 MSG("TOCZENIE GWINTU")
N15 T1 D1 G95 S600 F0.1 M4 M8  ; DO GWINTU
N20 G0 X55
N25 DIAMOF
N30 Z-45       ; I PRZEJSCIE
N35 G1 X20.2089
N40 G33 Z2 K4.5
N45 G0 X22.5
N50 Z-44.8258  ; II PRZEJSCIE
N55 G1 X19.8812
N60 G33 Z2 K4.5
N65 G0 X22.5
N70 Z-44.6921  ; III PRZEJSCIE
N75 G1 X19.6298
N80 G33 Z2 K4.5
N85 G0 X22.5
N90 Z-44.5794  ; IV PRZEJSCIE
N95 G1 X19.4178
N100 G33 Z2 K4.5
N105 G0 X22.5
N110 Z-44.4801  ; V PRZEJSCIE
N115 G1 X19.2311
N120 G33 Z2 K4.5
N125 G0 X22.5
N130 Z-44.3903   ; VI PRZEJSCIE
N135 G1 X19.0622
N140 G33 Z2 K4.5
N145 G0 X22.5
N150 Z-44.3077   ; VII PRZEJSCIE
N155 G1 X18.9070
N160 G33 Z2 K4.5
N165 G0 X22.5
N170 Z-44.2309   ; VIII PRZEJSCIE
N175 G1 X18.7625
N180 G33 Z2 K4.5
N185 G0 X22.5
N190 Z-44.1587   ; IX PRZEJSCIE
N195 G1 X18.6267
N200 G33 Z2 K4.5
N205 G0 X22.5
N210 Z-44.0905   ; X PRZEJSCIE
N215 G1 X18.4983
N220 G33 Z2 K4.5
N225 G0 X22.5
N230 Z-44.0256   ; XI PRZEJSCIE
N235 G1 X18.3762
N240 G33 Z2 K4.5
N245 G0 X22.5
N250 Z-43.9635   ; XII PRZEJSCIE
N255 G1 X18.2596
N260 G33 Z2 K4.5
```

```
N265 G0 X22.5
N270 Z-43.9040   ; XIII PRZEJSCIE
N275 G1 X18.1477
N280 G33 Z2 K4.5
N285 G0 X22.5
N290 Z-43.8468   ; XIV PRZEJSCIE
N295 G1 X18.0400
N300 G33 Z2 K4.5
N305 G0 X22.5
N310 Z-43.8468   ; PRZEJSCIE PUSTE
N315 G1 X18.0400
N320 G33 Z2 K4.5
N325 G0 X22.5
N330 DIAMON M0
N335 G53 T0 D0 G0 Z300 X300 M9 M5
N340 MSG("")
N345 M30
```



Rys. 133. Wynik symulacji obróbki gwintu M42

6.2. Interpolacja spiralna o zmiennym skoku G34/G35

W rzadko spotykanych przypadkach zachodzi konieczność nacięcia linii spiralnej o zmiennym skoku. Do tego celu służą funkcje G34 (rosnący skok) i G35 (malejący skok). Programowanie ruchu z funkcjami G34/G35 jest identyczne jak dla G33, dodatkowym parametrem jest zmiana skoku linii spiralnej ΔP, programowana pod adresem F w [mm/obr], np.

G34 Z20 K5 F0.01

G35 Z20 K5 F0.01

Wartość zmiany skoku gwintu można wyznaczyć z następującej zależności:

$$\Delta P = \frac{\left| p_b^2 - p_e^2 \right|}{2L_g - (p_b + p_e)} \tag{13}$$

gdzie: p_b – skok początkowy linii spiralnej;
p_e – skok końcowy linii spiralnej;
L_g – długość nacinanej linii spiralnej.

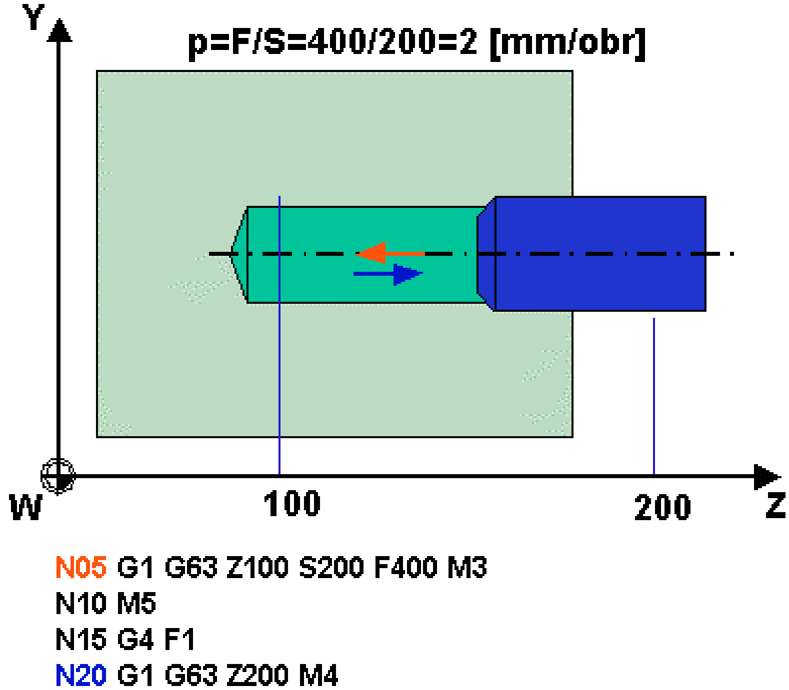
6.3. Nacinanie gwintów narzędziami kształtowymi bez kodera (G63)

Obróbka gwintów za pomocą narzędzi kształtowych (np. gwintowników) również wymaga skojarzenia ruchu obrotowego wrzeciona z posuwem liniowym. Takie narzędzie jak gwintownik niejako samoczynnie takie skojarzenie generuje, w zasadzie nie wymagając posuwu liniowego (obracany gwintownik sam "ciągnie"). Tym niemniej zaprogramowanie ruchu liniowego jest zawsze wymagane, stosuje się po prostu interpolację liniową G1. Sprzężenie posuwu liniowego z kątem obrotu wrzeciona jest uzyskiwane **wyłącznie** przez odpowiednie zaprogramowanie obrotów wrzeciona (adres S) i posuwu liniowego (adres F), bez stosowania dodatkowych urządzeń (kodera). Musi być zatem zachowany następujący warunek:

$$p = \frac{F}{S} \quad \frac{[\mathrm{mm/min}]}{[\mathrm{obr/min}]} = \frac{[\mathrm{mm}]}{[\mathrm{obr}]} \tag{14}$$

gdzie p jest skokiem gwintu. Zarówno wartość prędkości obrotowej wrzeciona, jak i posuwu może być modyfikowana przez operatora obrabiarki (ang. Override) za pomocą pokręteł lub przycisków na pulpicie maszynowym, co może doprowadzić do uzyskania innej wartości skoku niż zaprogramowana (a tym samym do zniszczenia narzędzia). Dlatego przewidziano niemodalną funkcję **G63** której zadaniem jest wyłączenie nastaw operatora, ustawiając je na wartości równe 100% dla adresów S i F, przywracając wartości zaprogramowane.

Gwintowanie z funkcją G63 (**w połączeniu z funkcją G1**) – Rys. 134 – z uwagi na brak dokładnego sprzężenia ruchu obrotowego z liniowym ruchem posuwu powinno być realizowane przy wykorzystaniu tzw. oprawek kompensacyjnych, umożliwiających korekcję osiową (wydłużanie i skracanie) narzędzia. Zadaniem programisty jest również zadbanie o odpowiednie kierunki obrotów wrzeciona (zarówno przy wgłębianiu jak i wycofywaniu narzędzia), stosownie do rodzaju narzędzia i samego gwintu. Po osiągnięciu zadanej głębokości gwintowania (przed zmianą kierunku obrotów wrzeciona) powinno zaprogramować się postój czasowy (funkcja G4 – patrz rozdz. 4).

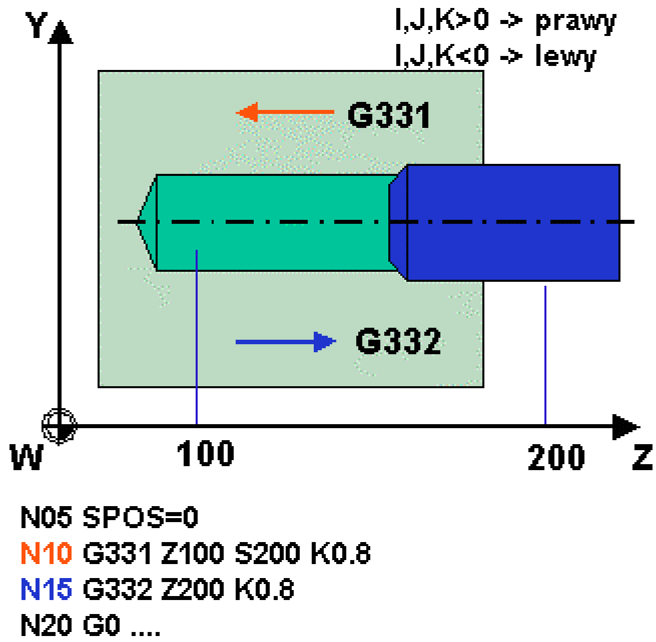


Rys. 134. Programowanie obróbki gwintu funkcją **G63**

6.4. Nacinanie gwintów narzędziami kształtowymi z koderem (G331/G332)

Interpolacja spiralna z wykorzystaniem funkcji G331/G332 przeznaczona jest do obróbki gwintów narzędziami kształtowymi (np. gwintownikami), przy czym w przeciwieństwie do obróbki z wykorzystaniem funkcji G63 wymaga dokładnego sprzężenienia ruchu obrotowego wrzeciona z liniowym ruchem posuwu. Jest ono realizowane przy użyciu tzw. kodera, tj. urządzenia do pomiaru położenia kątowego wrzeciona (wrzeciono traktowane jako oś sterowana numerycznie). Dlatego przy tym rodzaju interpolacji jest możliwa obróbka bez użycia oprawek kompensacyjnych.

Programując ruch z interpolacją **G331/G332** (podobnie jak w innych rodzajach interpolacji) zadaje się współrzędne punktu końcowego ruchu. Dodatkowo programuje się prędkość obrotową wrzeciona S podczas gwintowania, ponieważ przed rozpoczęciem gwintowania należy zaprogramować pozycjonowane zatrzymanie wrzeciona (funkcja SPOS – patrz rozdz. 4). Skok gwintu jest programowany pod parametrami interpolacji I, J, K stosownie do osi, wzdłuż której ruch się odbywa. Znak stojący przy wartości parametru interpolacji wpływa na kierunek obrotów wrzeciona przy nacinaniu gwintu (funkcja **G331**) – wartość dodatnia oznacza obroty prawe (M3), ujemna lewe (M4). Zakończenie ruchu związane jest z automatycznym zatrzymaniem obrotów wrzeciona. Przy wycofaniu narzędzia (funkcja **G332**) następuje uruchomienie obrotów wrzeciona w kierunku przeciwnym, jak przy funkcji G331. Znak parametru interpolacji musi być identyczny dla obu funkcji – Rys. 135.



Rys. 135. Programowanie obróbki gwintu przy użyciu interpolacji spiralnej G331/G332

6.5. Obróbka powierzchni spiralnych z użyciem funkcji G2/G3

Funkcje G2/G3 standardowo służą do programowania interpolacji kołowej w ustalonej płaszczyźnie układu współrzędnych. Gdyby jednak połączyć ruch po okręgu koła z ruchem liniowym w osi prostopadłej do płaszczyzny ruchu kołowego to wypadkową jest ruch po linii spiralnej (walcowej) – Rys. 120. Wykorzystanie interpolacji kołowej do obróbki powierzchni spiralnych ma miejsce przede wszystkim przy frezowaniu powierzchni o mniejszych wymaganiach (np. rowki smarowe).

Programowanie obróbki powierzchni spiralnych z użyciem interpolacji kołowej wymaga zadania dwóch ruchów w jednym bloku:

1. Ruchu po okręgu koła; wykorzystuje się następujące zestawy adresów (odpowiednio do płaszczyzny interpolacji G17/G18/G19):

➔ G2/G3 X... Y... Z... I... J... K...

➔ G2/G3 I... J... K... AR=...

➔ G2/G3 X... Y... Z... CR=...

2. Ruch liniowy w płaszczyźnie prostopadłej przy założeniu wykonania określonego przesunięcia w tej osi i zadanej liczby pełnych okręgów, programowanych pod adresem TURN oraz ewentualnego ruchu po łuku do punktu końcowego (jeżeli jego współrzędne w płaszczyźnie interpolacji są różne od współrzędnych punktu początkowego ruchu). Skok wykonanej spirali jest wynikowy, zależny od przebytej drogi kątowej (liczba okręgów) i liniowej (przesunięcie w osi dosuwowej).

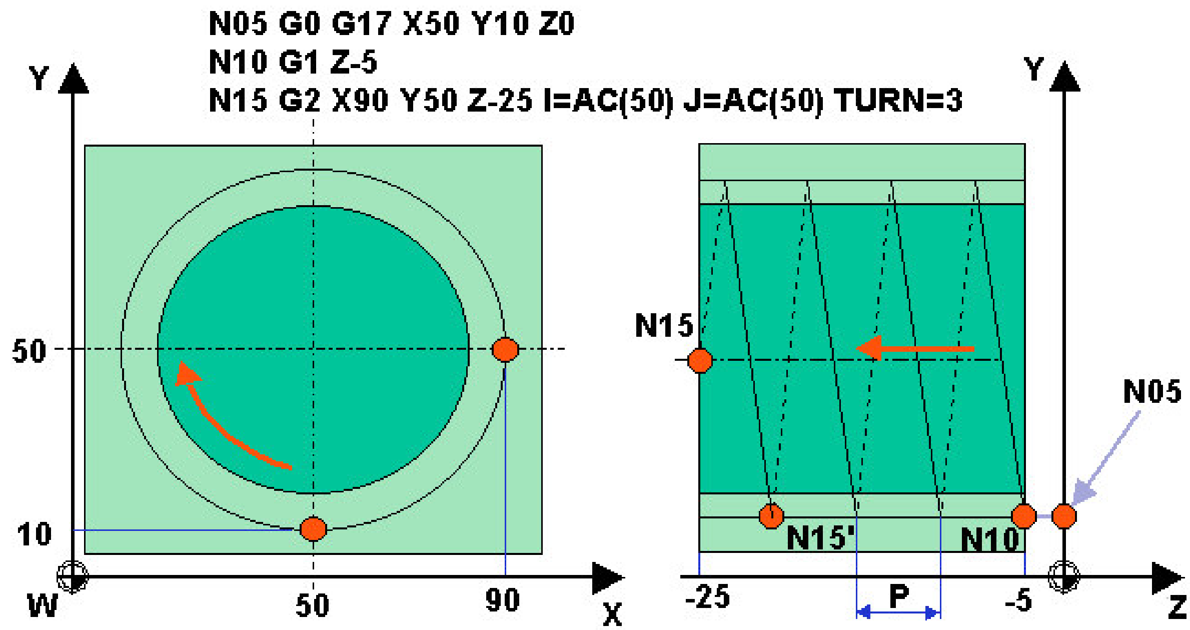
Przykład programowania ruchu z interpolacją spiralną i wykorzystaniem funkcji G2/G3 pokazano na Rys. 136. Przebieg obróbki jest następujący:

1. Ruch szybki na płaszczyznę bezpieczną (blok N05) z ustawieniem płaszczyzny interpolacji G17 (pł. XY).
2. Ruch roboczy dosuwowy do materiału (blok N10).

3. Ruch z interpolacją kołową w płaszczyźnie XY – zadany środek okręgu adresami I oraz J – z wykonaniem trzech pełnych okręgów (słowo TURN=3) do punktu o współrzędnych X i Y jak punkt początkowy (blok N15') – ruch w osi dosuwowej Z.
4. Ruch z interpolacją kołową do punktu o zadanych współrzędnych X i Y z wykonaniem ruchu po części łuku koła (w tym przypadku ¾ obwodu) z dalszym dosuwem w osi Z.

W tym przykładzie zostało zaprogramowanych łącznie 3 ¾ okręgów, z czego wynika wartość skoku spirali *P*, czyli odległość w osi Z pomiędzy początkiem a końcem ruchu podzielona przez liczbę okręgów:

$$P = \frac{-5-(-25)}{3.75} = \frac{20}{3.75} = 5.3333 \tag{15}$$



Rys. 136. Programowanie obróbki powierzchni spiralnej za pomocą funkcji **G2/G3**

7. INNE FUNKCJE PRZYGOTOWAWCZE

7.1. Postój czasowy

Postój czasowy to chwilowe wstrzymanie wykonywania kolejnych bloków programu (kolejnych ruchów narzędzia) przy zachowaniu aktualnego stanu obrabiarki. Jako przykłady sytuacji, kiedy takie zachowanie jest wymagane, można podać wstrzymanie posuwu przy wierceniu w celu złamania i usunięcia wiórów czy dokładne wytoczenie dna rowka przy toczeniu promieniowym.
Do programowania postoju czasowego służy niemodalna funkcja **G4**. Powinna ona występować jako jedyna funkcja przygotowawcza w bloku (nie można jej działania łączyć z wieloma innymi funkcjami, a przede wszystkim z ruchem narzędzia). Wartość postoju czasowego jest natomiast programowana na dwa sposoby:
G4 Fxx – wartość adresu F (xx) programuje czas postoju w [s];
G4 Sxx – wartość adresu S (xx) programuje czas postoju w [obr].

W pierwszym przypadku czas postoju programuje się bezpośrednio, w drugim zaś pośrednio – **wymaga się aby wrzeciona główne było włączone**. Wtedy rzeczywisty czas postoju zależy od jego prędkości obrotowej. Np. dla aktualnych obrotów 500 [obr/min] (G97 S500) zaprogramowanie G4 S20 oznacza postój czasowy równy:

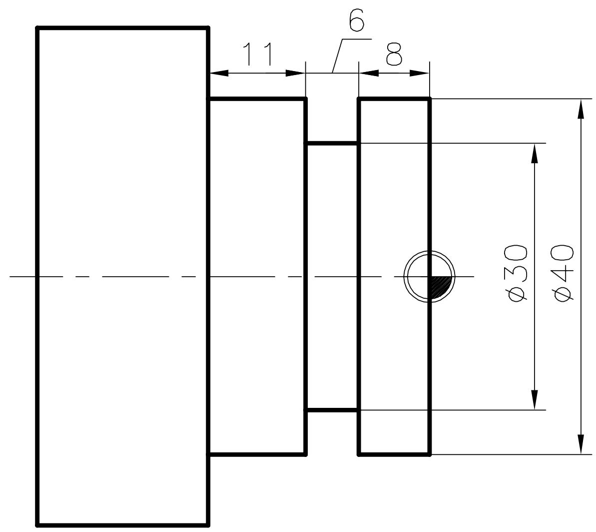
$$\frac{20\,[\text{obr}]}{500\,[\text{obr/min}]} = 0.04\,[\text{min}] = 2.4\,[\text{s}] \tag{16}$$

Uzależnienie postoju czasowego od obrotów wrzeciona jest **zalecanym sposobem**, zwłaszcza podczas obróbki. Wynika to faktu, że programując postój czasowy na ogół zakłada się pewną wartość prędkości obrotowej wrzeciona. Tym czasem wartość rzeczywista może zostać skorygowana przez operatora (najczęściej w przedziale 50÷120% wartości zaprogramowanej). To powoduje, że czas postoju w stosunku do bieżących obrotów jest za długi (przy zwiększonej prędkości), albo za krótki (przy zmniejszonej prędkości). Użycie adresu S zawsze będzie skutkowało stałym (w stosunku do aktualnych obrotów wrzeciona) wstrzymaniem wykonania programu.

Adresy S i F w połączeniu z funkcją G4 są **adresami niemodalnymi**, tzn. ich znaczenie podstawowe (prędkość obrotowa, posuw) nie ulega zmianie. Nie trzeba zatem w następnym bloku po postoju czasowym programować nowych ich wartości – automatycznie są przywracane ich poprzednie wartości, sprzed wywołania postoju czasowego

7.1.1. Przykład

Zaprogramować obróbkę rowka promieniowego za pomocą wcinaka o szerokości 4 mm – Rys. 137. Wykorzystać dwa rejestry, programujące lewe i prawe naroże.



Rys. 137. Szkic przedmiotu do przykładu z wykorzystaniem postoju czasowego

Rozwiązanie:

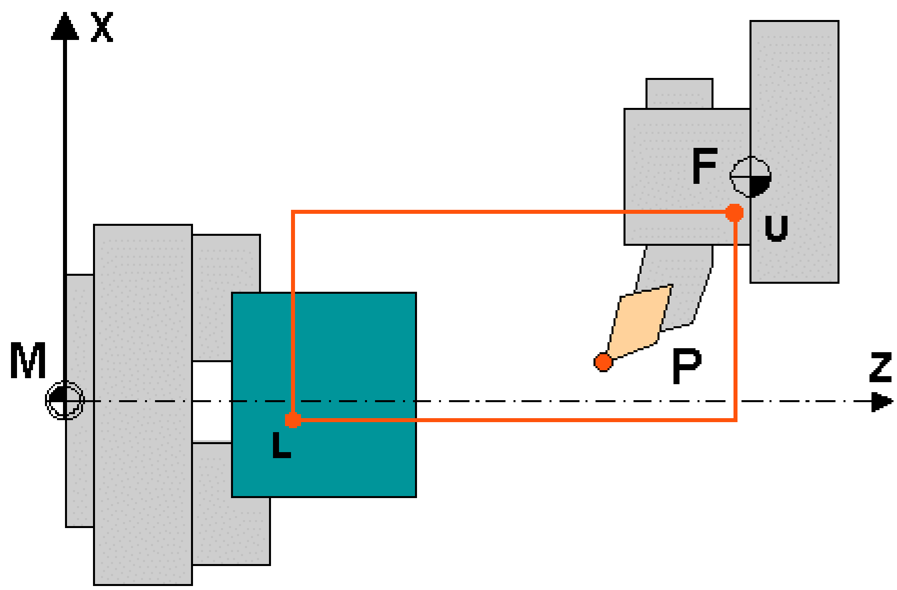
```
%_N_EX12_MPF
; 03-09-2003
N5 G40 G54 G71 G90 G96 DIAMON KONT G450
N10 MSG("TOCZENIE ROWKA")
N15 T2 D1 S125 F0.12 M4 M8 ;LEWE NAROZE
N20 G0 X50
N25 Z-14
N30 X44
N35 G1 X30
N40 G4 S8 ;POSTOJ PRZEZ 8 OBROTOW
N45 G0 X42
N50 D2              ;PRAWE NAROZE
N55 Z-8
N60 G1 X30
N65 G4 S8
N70 G0 X50
N75 MSG("")
N80 G53 T0 D0 G0 Z300 X300 M9 M5
N85 M30
```

7.2. Obszary robocze

W przestrzeni roboczej każdej obrabiarki znajduje się nie tylko narzędzie i przedmiot obrabiany, ale szereg innych obiektów (oprawki narzędziowe, uchwyty, oprzyrządowanie itp.) Wszystkie są potencjalnym źródłem kolizji, tj. kontaktu z ruchomymi zespołami obrabiarki, poruszającymi się ruchem szybkim. Każda kolizja niesie potencjalne ryzyko uszkodzenia elementu uczestniczącego w niej, stąd jednym z podstawowych celów weryfikacji programu sterującego jest uniknięcie takich sytuacji. Niestety, programista nie zawsze jest w stanie przewidzieć rozmieszczenia wszystkich obiektów w przestrzeni roboczej, a tym samym i źródeł kolizji.

Zapobiegać sytuacjom kolizyjnym mają różne zabezpieczenia fizycznie zamontowane na obrabiarce (np. wyłączniki krańcowe), jak również tzw. **bariery elektroniczne**. Są to **zadawane z pulpitu sterującego** wartości współrzędnych

w układzie maszynowym (MKS), które na danym egzemplarzu obrabiarki nie mogą zostać przekroczone (ani z poziomu obsługi ręcznej, ani z poziomu programu sterującego). Wyznaczają one obszar dopuszczalnych położeń (inaczej obszar roboczy) punktu kodowego narzędzia **P** (lub w przypadku jego braku punktu kodowego **F**). Na Rys. 138 pokazano przykład obszaru roboczego dla tokarki (czerwony prostokąt), definiowanego przez dolne (punkt **L**) i górne (punkt **U**) współrzędne graniczne. W przypadku frezarek obszar roboczy przyjmuje postać prostopadłościanu.



Rys. 138. Przykład obszaru roboczego dla tokarki

Z poziomu programu sterującego jest możliwe programowanie zakresu obszaru chronionego (choć nie jest to zalecane) przy pomocy dwóch niemodalnych funkcji przygotowawczych (zadawanych w osobnych blokach):

G25 X... Y... Z.... – dolne ograniczenie obszaru roboczego;
G26 X... Y... Z.... – górne ograniczenie obszaru roboczego.

Występujące razem z funkcjami współrzędne podają współrzędne graniczne, nie łączą się zatem z programowaniem ruchu. Należy mieć na uwadze, że zaprogramowane wartości **są zapamiętywane w układzie sterowania i obowiązują również przy wykonywaniu innych programów**. Z programową kontrolą obszaru roboczego są związane dwie kolejne modalne funkcje przygotowawcze:

WALIMON – (ang. Work Area LIMitation ON), włączenie programowego ograniczania obszaru roboczego;
WALIMOF – (ang. Work Area LIMitation OFf), wyłączenie programowego ograniczania obszaru roboczego.

Zatem ograniczanie zakresu dopuszczalnych położeń narzędzia jest uwzględniane przez układ sterowania tylko przy aktywnej funkcji WALIMON.

7.3. Najazd na punkt referencyjny

Najazd na punkt referencyjny (**R**) zasadniczo wykonywany jest w trybie ręcznym po włączeniu obrabiarki. Tym niemniej istnieje funkcja przygotowawcza G74 (rzadko wykorzystywana) pozwalająca z poziomu programu wykonać ruch szybki zespołów obrabiarki na punkt referencyjny:

G74 X1=0 Y1=0 Z1=0

Współrzędne X1, Y1,... oznaczają osie maszynowe w których ma się odbyć ruch. Wartości współrzędnych nie mają tu znaczenia, najczęściej przyjmuje się zera. Powyższą funkcję można wykorzystać do ustawiania zespołów obrabiarki w ustalonym położeniu po zakończeniu programu, które stanowi jednocześnie punkt startowy dla kolejnego programu. Przyjęcie zasady, iż jest to zawsze punkt referencyjny uniezależnia współrzędne takiego punktu od obrabiarki na której program jest wykonywany (nie trzeba zadawać tych współrzędnych bezpośrednio w programie).

7.4. Najazd na punkt stały

W trakcie wykonywania programu sterującego zdarzają się sytuacje, kiedy zespoły obrabiarki powinny znaleźć się w pewnym specyficznym punkcie, współrzędne którego zna operator, nie zna natomiast programista. Może to być punkt wymiany narzędzia, wymiany palety narzędziowej, wykonania czynności pomiarowych itp. W tym celu można zapisać współrzędne pewnych punktów charakterystycznych w układzie sterowania, natomiast w programie sterującym nakazać wykonanie najazdu na punkt o oznaczeniu kodowym, którego rzeczywiste współrzędne pobrane zostaną z pamięci układu (dla układów sterowania Sinumerik 810D/840D współrzędne przechowywane są pod zmienną $MA_FIX_POS). Najazd na punkt stały programuje niemodalna funkcja przygotowawcza G75:

G75 FP=... X1=0 Y1=0 Z1=0

gdzie adres FP (ang. *Fixed Point*) programuje numer kolejny punktu stałego, przechowywany w układzie sterowania (dozwolone są numery 1, 2, ...). Współrzędne X1, Y1,..., podobnie jak dla funkcji G74 oznaczają osie maszynowe w których ma się odbyć ruch (wartości adresów bez znaczenia).

7.5. Sterowanie położeniem kątowym wrzeciona

Wrzeciono pracuje głównie w trybie dyskretnym (włącz/wyłącz). Coraz częściej spotyka się jednak łączenie wielu różnych sposobów obróbki na jednej obrabiarce, a to wymaga sterowania położeniem kątowym wrzeciona (np. frezowanie powierzchni nieobrotowych na centrach tokarskich). Formalnie napęd wrzeciona stanowi oś sterowaną numerycznie (dla wrzeciona głównego jest to oś C), jednak ze względu na mieszany charakter pracy stosuje się nieco inne rozwiązanie – wrzeciono może pracować w dwóch trybach:

- ➔ sterowania prędkością obrotową (wrzeciono pracuje jako napęd główny);
- ➔ sterowania położeniem (wrzeciono pracuje jako element pozycjonowany, rolę napędu głównego przejmuje inne wrzeciono, np. narzędzi napędzanych w centrum tokarskim).

Pierwszy z wymienionych trybów obsługują omówione wcześniej funkcje M3, M4, M5 – rozdz. 2.7. Do sterowania w drugim trybie służy funkcja **SPOS=....**

(ang. *Spindle POSition*), której wartość wyrażona jest w stopniach (0÷360°). Funkcja ta automatycznie przełącza wrzeciono w tryb sterowania położeniem (o ile wcześniej to nie zostało zrobione). Jeżeli wrzeciono jest w ruchu obrotowym to najpierw następuje jego zatrzymanie, a potem ustawienie w zadanej pozycji kątowej. Kąt zero na ogół jest ustalony w jakimś charakterystycznym położeniu, np. dla tokarki jest to górne pionowe ustawienie szczęki nr 1 uchwytu samocentrującego (zawsze trzeba to jednak sprawdzić).

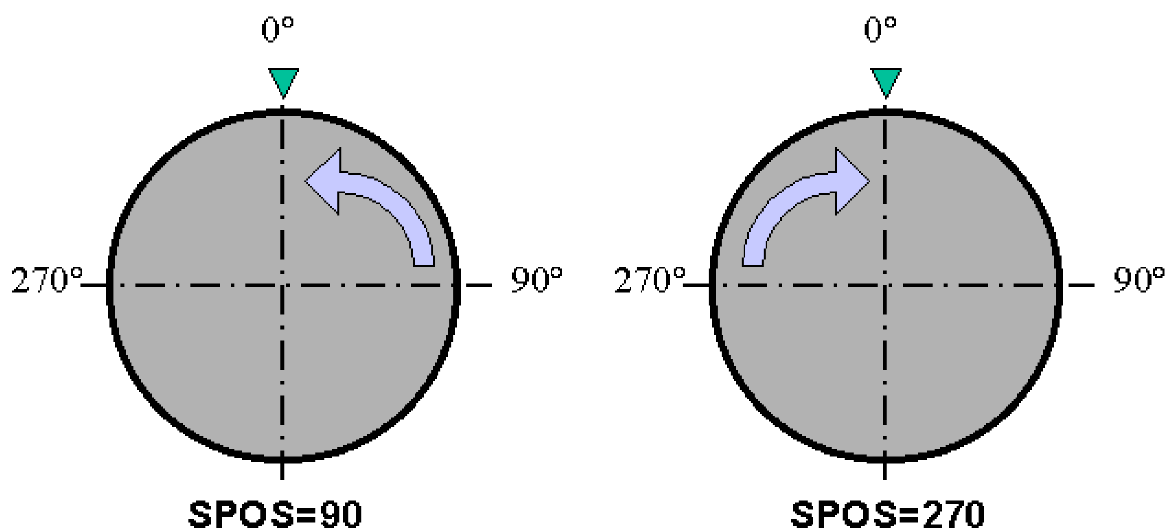
Przy zmianie położenia kątowego wrzeciona ruch odbywa się zawsze w takim kierunku, aby jego droga kątowa była najkrótsza – Rys. 139. Można narzucić kierunek pozycjonowania, jeżeli użyje się niemodalnych funkcji **ACP** (ang. *Angle Coordinate Positive*) – pozycjonowanie w kierunku dodatnim i **ACN** (ang. *Angle Coordinate Negative*) – pozycjonowanie w kierunku ujemnym – Rys. 140.

Do sterowania trybem pracy wrzeciona służą dwie modalne funkcje przygotowawcze:

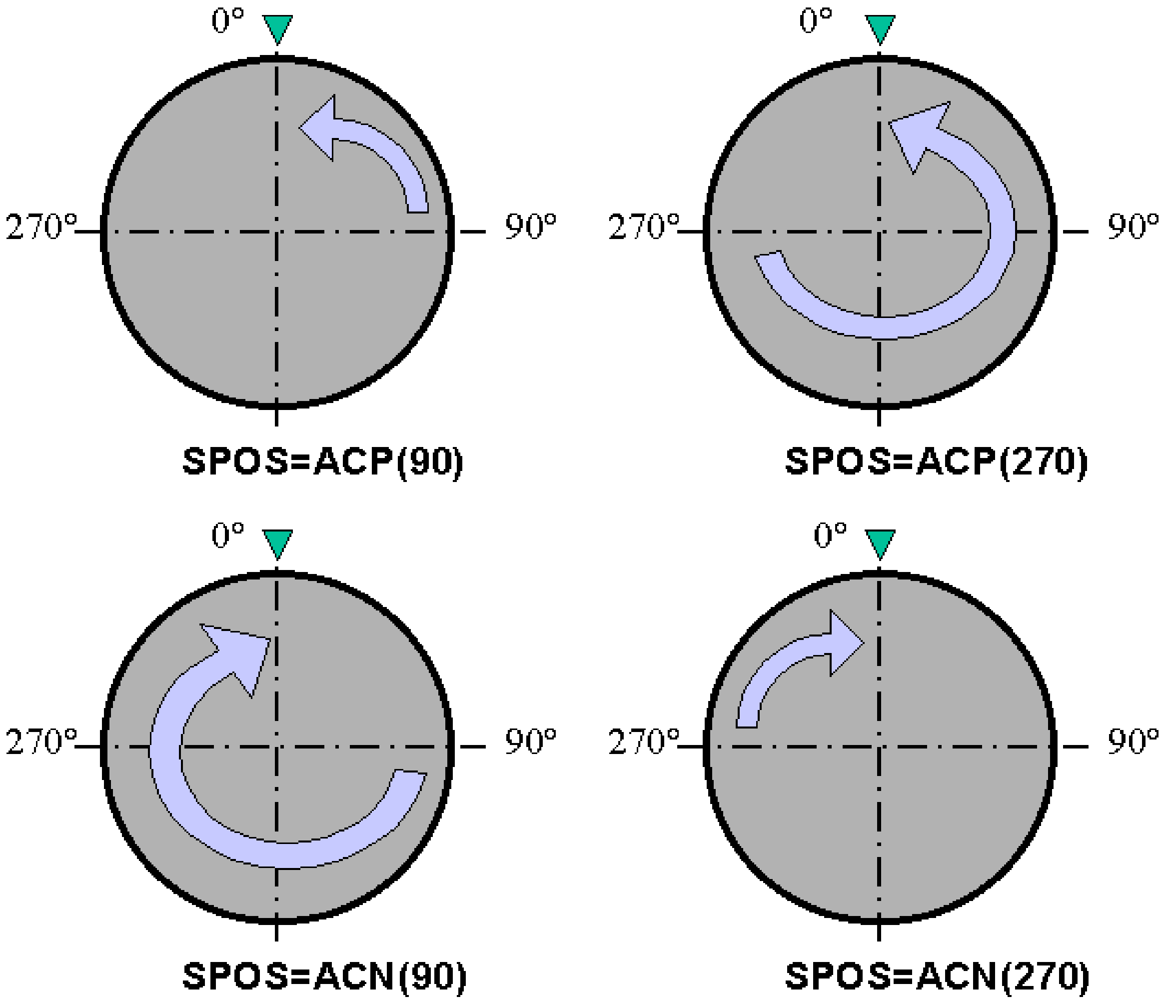
SPCON – (ang. *SPeed Control ON*) włączenie sterowania położeniem kątowym wrzeciona;

SPCOF – (ang. *Speed Control OFf*) włączenie sterowania prędkością obrotową wrzeciona.

Przykład wykorzystania pozycjonowania kątowego wrzeciona w obróbce przedmiotu na centrum tokarskim zamieszczono w rozdz. 5.3.2.



Rys. 139. Kierunki pozycjonowania wrzeciona przy ustawieniu początkowym 0° (patrząc od przodu wrzeciona)



Rys. 140. Kierunki pozycjonowania wrzeciona przy ustawieniu początkowym 0° przy wymuszeniu kierunku pozycjonowania

7.6. Sterowanie dokładnością ruchu narzędzia

(W przygotowaniu)

8. Programowanie parametryczne

Programowanie kojarzy się często z algorytmami, wykonywaniem obliczeń itp. Okazuje się, że w programach sterujących obrabiarkami czy innymi maszynami sterowanymi numerycznie te elementy również mogą się pojawić. W niniejszym rozdziale przedstawiono podstawowe wiadomości na ten temat, pozwalające jednak na tworzenie dość skomplikowanych programów.

8.1. R-parametry

Chcąc wykonywać jakiekolwiek obliczenia w programie komputerowym (również w sterującym) musimy wykorzystywane wartości gdzieś przechowywać. Miejsce w pamięci komputera, gdzie taka wartość jest przechowywana, jest adresowane, a adres ten najczęściej nazywamy **zmienną** (ang. *Variable*). Najczęściej używane zmienne pozwalają na przechowywanie liczb rzeczywistych, całkowitych, tablic, napisów tekstowych itp. Aby użyć zmiennej trzeba ją najpierw zadeklarować, aby komputer wiedział co dana zmienna reprezentuje (np. w Pascalu instrukcja **var**). W Sinumeriku również istnieje możliwość używania zmiennych o dowolnej nazwie i typie, np. bloki:

```
DEF REAL DIM
DEF INT A,B
DEF STRING LAN
DEF AXIS X,Y,Z
```

deklarują zmienną rzeczywistą DIM, całkowite A i B, łańcuchową LAN i osiowe (specjalny typ) X, Y i Z. Cechą charakterystyczną takich zmiennych jest ich lokalność (dostępne tylko w obrębie programu, w którym są zadeklarowane). Ich stosowanie zaleca się jednak dla bardziej doświadczonych programistów. Na poziomie średnio zaawansowanym do dyspozycji programisty jest narzędzie o nieco mniejszych możliwościach. Są to tzw. **R-parametry**.

R-parametry to jednowymiarowa tablica w pamięci układu sterowania, indeksowana adresami R (w niektórych układach sterowania mogą to być inne adresy, np. Q). W tablicy tej przechowywane są liczby rzeczywiste. Liczba komórek tablicy, a więc i zakres R-parametrów zależy od konkretnego układu, w Sinumeriku jest to 1000 adresów od R0 do R999. Odwołując się do *x*-tej komórki tablicy przez adres R*x* można zarówno pobierać z niej dane, jak i tam je wprowadzać. Zaletą R-parametrów jest to, że podgląd ich wartości jest dostępny na pulpicie operatorskim. Możliwe jest zatem śledzenie zmian ich wartości w trakcie wykonywania programów sterujących, co pozwala m.in. na diagnostykę błędów.

Do czego mogą służyć R-parametry? Przede wszystkim ich wartości mogą być przypisywane do innych adresów, o czym wspomniano w rozdz. 2.1. Poza tym ich przeznaczeniem może być przechowywanie informacji zbieranych w trakcie wykonywania programów sterujących. Możliwe jest zliczanie wykonanych sztuk przedmiotów, zliczanie czasu pracy narzędzi itp.

8.2. Obliczenia na R-parametrach

Podstawowym celem stosowania R-parametrów jest wykonywanie na nich obliczeń. Mogą to być zarówno proste operacje arytmetyczne, jak i skomplikowane funkcje trygonometryczne, logarytmiczne i inne (można na nich operować również wartościami stałymi). Najważniejsze operatory i funkcje dostępne w Sinumeriku 840D to:

+	– operator dodawania;
-	– operator odejmowania;
*	– operator mnożenia;
/	– operator dzielenia;
(,)	– nawiasy, grupowanie wyrażeń, argumenty funkcji;
DIV()	– dzielenie całkowitoliczbowe;
MOD	– dzielenie modulo;
SIN()	– funkcja sinus (argument w [º]);
COS()	– funkcja cosinus (argument w [º]);
TAN()	– funkcja tangens (argument w [º]);
ASIN()	– odwrotna funkcja sinus (wynik w [º]);
ACOS(),	– odwrotna funkcja cosinus (wynik w [º]);
ATAN2(,)	– odwrotna funkcja tangens (wynik w [º]);
SQRT()	– pierwiastek kwadratowy;
ABS()	– moduł liczby;
POT()	– kwadrat liczby (potęga o wykładniku 2);
TRUNC()	– część całkowita liczby;
ROUND()	– zaokrąglenie do liczby całkowitej;
LN()	– logarytm naturalny;
EXP()	– funkcja ekspotencjalna (e^x).

Przykłady:

R10=2	przypisanie wartości 2 do R-parametru R10
R12=5	
R13=7	
R14=2	inne operacje przypisania
R10=R12	przepisanie wartości z R12 do R10 (R10=5)
R10=R12+2	operacja dodawania wartości z R12 i stałej 2 (R10=7)
R10=R12+R13-R14	obliczenia dodawania i odejmowania (R10=10)
R10=R12*R13/R14	obliczenia mnożenia i dzielenia (R10=17.5)
R10=R14+R12*R13	obliczenia mnożenia i dodawania (R10=37)

W tym ostatnim przykładzie najpierw wykonano mnożenie (operatory mnożenia i dzielenia mają wyższy priorytet niż dodawania i odejmowania) a potem dodawania. Jeżeli chcemy jasno sprecyzować kolejność wykonywania obliczeń można użyć nawiasów, np.

R10=(R14+R12)*R13	R10=49
R10=12 DIV 10	R10=1
R10=12 MOD 2	R10=2
R10=SIN(30)	R10=0.5

R10=COS(30)	R10=0.866
R10=TAN(30)	R10=0.5773
R10=ASIN(0.4)	R10=23.578
R10=ACOS(0.4)	R10=66.422
R10=ATAN2(0.4,1)	R10=21.801

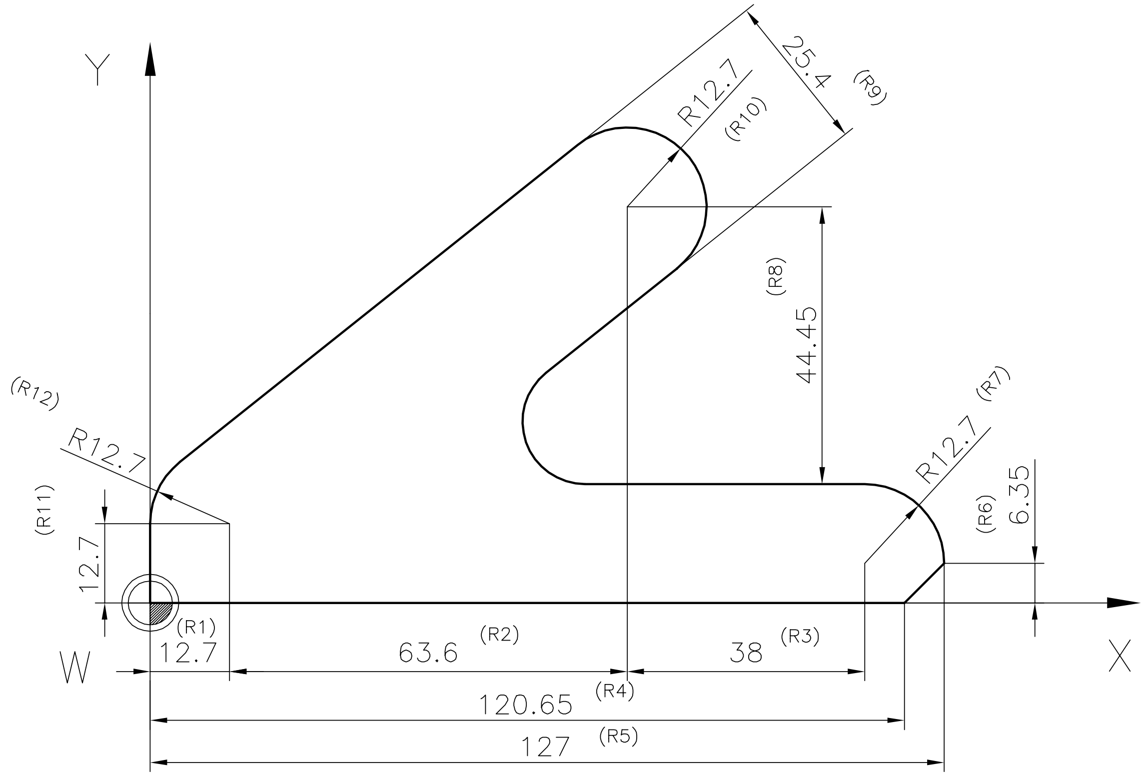
Funkcja odwrotna do tangens jest funkcją dwuargumentową. W pierwszej kolejności pierwszy argument jest dzielony przez drugi, a dopiero na wyniku dzielenia jest obliczana funkcja. Można zatem powiedzieć, że argumenty funkcji to długości przyprostokątnych trójkąta prostokątnego, dla którego obliczamy kąt między przeciwprostokątną a drugą przyprostokątną.

R10=SQRT(7)	R10=2.645
R10=ABS(-7.8)	R10=7.8
R10=POT(3.5)	R10=12.25
R10=TRUNC(3.5)	R10=3
R10=TRUNC(-3.5)	R10=-3
R10=ROUND(3.4)	R10=3
R10=ROUND(3.5)	R10=4
R10=ROUND(-3.5)	R10=-4
R10=LN(3.5)	R10=1.252
R10=EXP(3.5)	R10=33.115

Przy obliczaniu wartości na R-parametrach należy pamiętać o przedziałach określoności funkcji, np. nie liczyć pierwiastka z liczby ujemnej. Obliczając bardziej skomplikowane wyrażenia zaleca się rozbić je na prostsze podwyrażenia, a dopiero później łączyć w większe całości. Ułatwia to znacznie wykrywanie ewentualnych błędów.

8.2.1. Przykład

Zaprogramować obróbkę konturu (od zewnątrz) przedstawionego na Rys. 141. Nieznane wymiary obliczyć korzystając z R-parametrów. Przyjąć powierzchnię materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 6 mm. Rozpocząć od punktu (0,0) w kierunku przeciwnym do ruchu wskazówek zegara.



Rys. 141. Szkic przedmiotu do przykładu obróbki z wykorzystaniem obliczeń na R-parametrach

Rozwiązanie:

Do poprawnego zaprogramowania obróbki koniecznym jest wyznaczenie współrzędnych czterech punktów – P1, P2, P3 i P4 (Rys. 142). Do obliczeń przyjęto, że każdy z wymiarów podanych na rysunku konstrukcyjnym będzie zapisany w programie pod oddzielnym R-parametrem. Adresy tych R-parametrów podano na Rys. 141 (oznaczenia w nawiasach). Kolejność obliczeń jest następująca (w nawiasach podano adresy R-parametrów, pod którym dany wynik będzie przechowany w programie sterującym) (Rys. 142):

$$\frac{\overline{BD}}{\overline{AD}} = \mathrm{tg}\,\alpha \quad \Rightarrow \quad \alpha = \mathrm{arctg}\frac{\overline{BD}}{\overline{AD}} = \mathrm{arctg}\frac{\mathrm{R6}+\mathrm{R7}+\mathrm{R8}-\mathrm{R11}}{\mathrm{R2}} \quad (\mathrm{R13})$$

$$\frac{\overline{CP3}}{\overline{BP3}} = \sin\alpha \quad \Rightarrow \quad \overline{CP3} = \overline{BP3}\cdot\sin\alpha = \mathrm{R10}\cdot\sin(\mathrm{R13}) \quad (\mathrm{R14})$$

$$\frac{\overline{BC}}{\overline{BP3}} = \cos\alpha \quad \Rightarrow \quad \overline{BC} = \overline{BP3}\cdot\cos\alpha = \mathrm{R10}\cdot\cos(\mathrm{R13}) \quad (\mathrm{R15})$$

$$\overline{BE} = \overline{BC} = \mathrm{R15}, \quad \overline{EP2} = \overline{CP3} = \mathrm{R14}$$

$$X_{P3} = X_B + \overline{CP3} = \mathrm{R1}+\mathrm{R2}+\mathrm{R14} \quad (\mathrm{R16})$$

$$X_{P2} = X_B - \overline{CP3} = \mathrm{R1}+\mathrm{R2}-\mathrm{R14} \quad (\mathrm{R17})$$

$$Y_{P3} = Y_B - \overline{BC} = \mathrm{R6}+\mathrm{R7}+\mathrm{R8}-\mathrm{R15} \quad (\mathrm{R18})$$

$$Y_{P2} = Y_B + \overline{BC} = \mathrm{R6}+\mathrm{R7}+\mathrm{R8}+\mathrm{R15} \quad (\mathrm{R19})$$

$$\overline{P3F} = Y_B - \overline{BC} - \overline{FG} = \text{R8} - \text{R15} \quad \text{(R20)}$$

$$\frac{\overline{P3F}}{\overline{FP4}} = \text{tg}\,\alpha \quad \Rightarrow \quad \overline{FP4} = \frac{\overline{P3F}}{\text{tg}\,\alpha} = \frac{\text{R20}}{\text{tg(R13)}} \quad \text{(R21)}$$

$$X_{P4} = X_{P3} - \overline{FP4} = \text{R16} - \text{R21} \quad \text{(R22)}$$

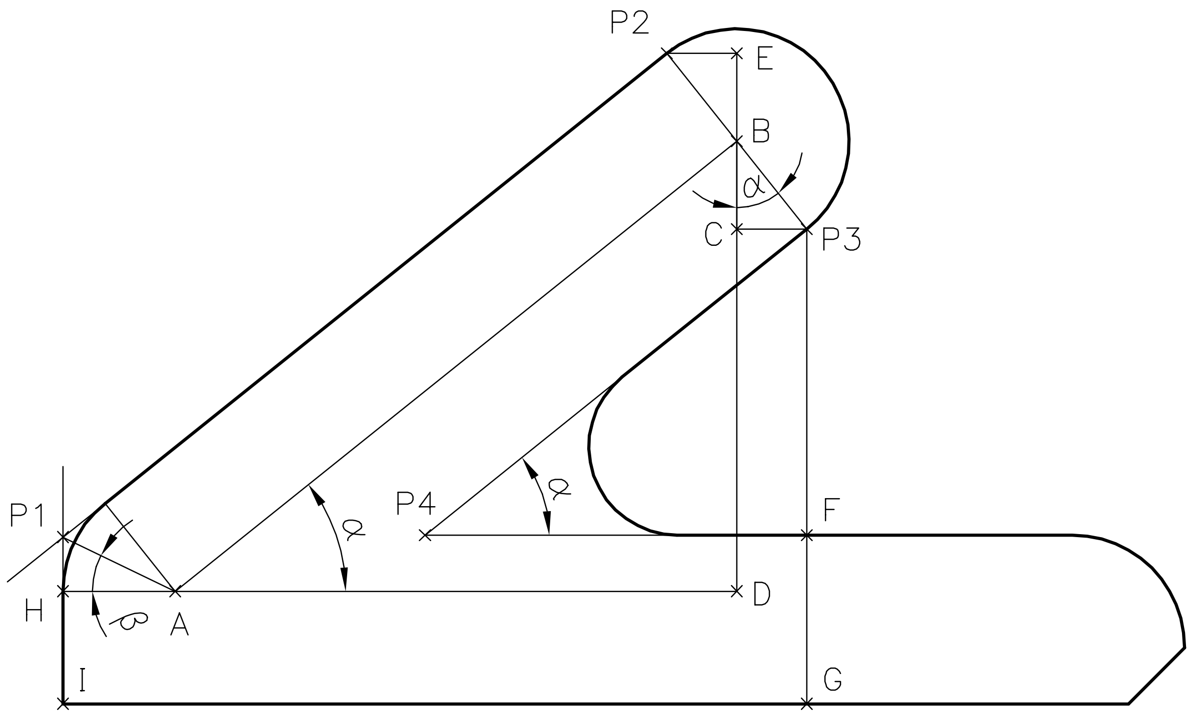
$$Y_{P4} = \text{R6} + \text{R7} \quad \text{(R23)}$$

$$\beta = \frac{180 - 90 - \alpha}{2} = \frac{90 - \text{R13}}{2} \quad \text{(R24)}$$

$$\frac{\overline{HP1}}{\overline{AH}} = \text{tg}\,\beta \quad \Rightarrow \quad \overline{HP1} = \overline{AH} \cdot \text{tg}\,\beta = \text{R1} \cdot \text{tg(R24)} \quad \text{(R25)}$$

$$X_{P1} = 0 \quad \text{(R26)}$$

$$Y_{P1} = \overline{IH} + \overline{HP1} = \text{R11} + \text{R25} \quad \text{(R27)}$$



Rys. 142. Poszukiwane punkty i zasady wyznaczania ich współrzędnych

Program sterujący:

```
%_N_EX09_MPF
; 25-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
MSG("OBLICZENIA")
; DANE WEJSCIOWE
R1=12.7
R2=63.6
R3=38
R4=120.65
R5=127
R6=6.35
R7=12.7
R8=44.45
```

```
R9=25.4
R10=12.7
R11=12.7
R12=12.7
; DANE POSREDNIE I WYJSCIOWE
R13=ATAN2(R6+R7+R8-R11,R2)
R14=R10*SIN(R13)
R15=R10*COS(R13)
R16=R1+R2+R14
R17=R1+R2-R14
R18=R6+R7+R8-R15
R19=R6+R7+R8+R15
R20=R8-R15
R21=R20/TAN(R13)
R22=R16-R21
R23=R6+R7
R24=(90-R13)/2
R25=R1*TAN(R24)
R26=0
R27=R11+R25
MSG("OBROBKA")
N10 T1 D1 S1500 F250 M3 M8 M6
N15 G0 X-20 Y-10
N20 Z3
N25 G1 Z-5
N30 G42 X0 Y0
N35 X=R4
X=R5 Y=R6
G3 I=-R7 AR=90
G1 X=R22 RND=10  ; DO PUNKTU P4
X=R16 Y=R18      ; DO PUNKTU P3
G3 X=R17 Y=R19 CR=R10  ; DO PUNKTU P2
G1 X=R26 Y=R27 RND=R12 ; DO PUNKTU P1
X0 Y0
G0 G40 Y-20
N40 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
MSG("")
N45 M30
```

8.3. Instrukcje strukturalne

Wszystkie dotąd zaprezentowane w przykładach programy sterujące miały jedną wspólną cechę: bloki były wykonywane w takiej kolejności, w jakiej były zapisane w programie. Podobnie jak ma to miejsce w tradycyjnym programowaniu możliwe są inne sposoby realizacji bloków programu sterującego: rozgałęzienia, pętle itp., realizowane za pomocą tzw. instrukcji strukturalnych. W języku układu sterowania Sinumerik 840D jest ich wiele, w niniejszym skrypcie zostaną omówione tylko te najważniejsze, stosowane przez średnio zaawansowanych programistów.

Do podstawowych instrukcji strukturalnych należy **instrukcja warunkowa IF.** Jej działanie w najprostszej postaci polega na sprawdzaniu pewnego warunku logicznego. Jeżeli warunek ten jest prawdziwy, to jest kontynuowana dalsza analiza bieżącego bloku. W przeciwnym przypadku analiza ta jest przerywana i następuje przejście do kolejnego bloku.

Warunek logiczny to najczęściej porównanie dwóch wartości liczbowych. Do operatorów porównania w Sinumeriku zalicza się:

== – równy (dwa znaki =);
<> – różny;
> – większy;
>= – większy lub równy;
< – mniejszy;
<= – mniejszy lub równy.

Przykłady:

IF R10>R11 R12=R10-R11 jeżeli R10>R11 to R12=R10-R11
R10=R12-R13
IF R10 R12=R13*R14 jeżeli R10>0 to R12=R13*R14

Jeżeli lista instrukcji, jaka ma być wykonana w przypadku pozytywnego lub negatywnego warunku logicznego jest długa, to może zawierać się w kilku blokach, jednak wtedy trzeba użyć bardziej rozbudowane postaci instrukcji warunkowej: **IF ENDIF,** przy czym adresy IF i ENDIF muszą być jedynymi adresami w bloku.

Przykład:

```
IF R10>R11
    R12=R13+1
    R14=R12-8
ENDIF
R15=R16*R18
```

W przypadku pozytywnej weryfikacji warunku R10>R11 wykonywane są dwa następne bloki. W przeciwnym przypadku dalsza analiza programu rozpoczyna się od piątego bloku (R15=R16*R18).

Jeszcze bardziej rozbudowana postać instrukcji warunkowej obejmuje operator **ELSE**, tzn. zakres bloków, jakie są wykonywane w przypadku negatywnej weryfikacji warunku logicznego.

Przykład:

```
IF R10>R11
    R12=R13+1
    R14=R12-8
ELSE
    R17=R14+R15
    R16=R20-R19
ENDIF
R15=R16*R18
```

W przypadku negatywnej weryfikacji warunku R10>R11 wykonywany jest ciąg bloków między instrukcjami ELSE i ENDIF. Niezależnie od warunku logicznego kolejnym wykonywanym blokiem jest blok ostatni (R15=R16*R18). Dla ułatwienia

analizy programu można stosować wcięcia dla bloków znajdujących się w obrębie instrukcji IF ELSE ENDIF.

Drugą grupą funkcji strukturalnych są **skoki bezwarunkowe**. Ich działanie polega na ominięciu dalszych adresów w bieżącym bloku i przejście do analizy i wykonania innego bloku (odpowiednik funkcji GOTO w Pascalu). W Sinumeriku istnieją dwie funkcje skoków bezwarunkowych:

GOTOB LABEL – skok bezwarunkowy w tył (ang. *GOTO Back*) do etykiety LABEL;

GOTOF LABEL – skok bezwarunkowy w przód (ang. *GOTO Forward*) do etykiety LABEL.

Przy skoku w tył jest poszukiwana etykiety od bieżącego bloku w kierunku początku programu, przy skoku w przód – w kierunku końca programu. Poszukiwanie kończy się z chwilą znalezienia bloku zawierającego podaną w instrukcji etykietę (etykiety omówiono również w rozdz. 2.8). Oznacza to, że w programie sterującym może być kilka bloków z tą samą etykietą a mimo tego adres skoku jest wyznaczony w sposób jednoznaczny. Jeżeli przy przeszukiwaniu programu układ sterowania nie znajdzie podanej etykiety to przerywa wykonanie programu sygnalizując błąd.

Przykład:

```
N5 G1 X0 Y0
ETYK1: G1 Y100
N10 X80
N15 X20 Y120
N20 GOTOB ETYK1
N25 G0 X200 Y200
```

W tym przykładzie pokazano pętlę, która nigdy się nie zakończy (a więc jest to błąd), polegającą na wykonywaniu skoków z bloku N20 zawsze do bloku o etykiecie ETYK1. Aby przerwać wykonywanie pętli należy zastosować **skoki warunkowe**. Są one połączeniem instrukcji GOTOB/GOTOF i poznanej wcześniej instrukcji warunkowej IF. Mają następującą postać:

IF <warunek >GOTOB LABEL – skok warunkowy w tył (ang. *GOTO Back*) do etykiety LABEL;

IF <warunek> GOTOF LABEL – skok warunkowy w przód (ang. *GOTO Forward*) do etykiety LABEL.

Przed wykonaniem skoku jest sprawdzany warunek logiczny i dopiero jego pozytywna weryfikacja pozwala na wykonanie skoku (w przód lub w tył) do bloku z zadaną etykietą.

Przykład:

```
N5 G1 X0 Y0 R1=0
ETYK1: G1 Y100
N10 X80 R1=R1+1
N15 X20 Y120
N20 IF R1<10 GOTOB ETYK1
N25 G0 X200 Y200
```

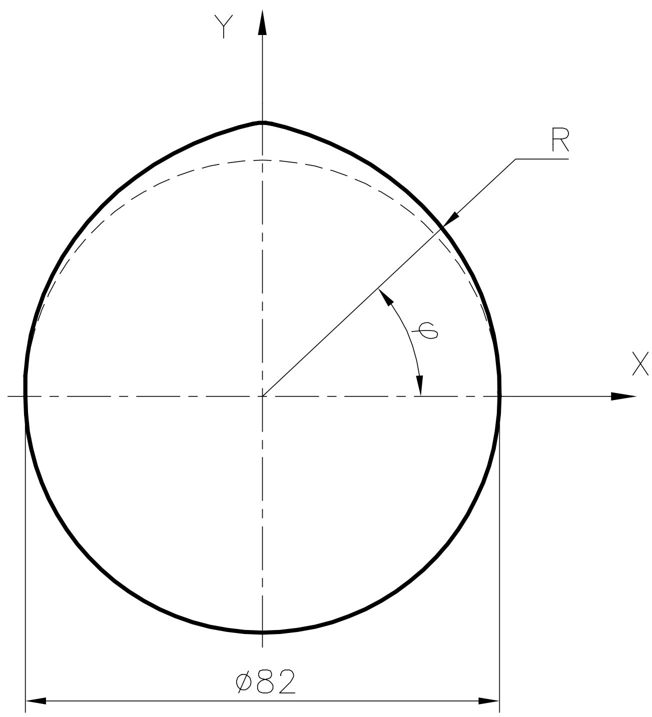
W powyższym przykładzie na początku zainicjowaną zmienną indeksującą (R1=0), którą przy każdym wykonaniu pętli zwiększa się o 1 (R1=R1+1). Sama pętla jest

wykonywana do czasu, kiedy zmienna indeksująca osiągnie wartość 10. Tym samym przy pomocy instrukcji skoku warunkowego można skonstruować inne, znane z języków strukturalnych, instrukcje pętli warunkowych **REPEAT UNTIL** czy **WHILE DO**.

Instrukcje skoków warunkowych pozwalają na dużą elastyczność w sterowaniu przebiegiem analizy i wykonania programów sterujących. Mogą one być pomocne w programowaniu np. przedmiotów o podobnych kształtach lecz różnych wymiarach, czy w przygotowaniu podprogramów obróbki pewnych fragmentów przedmiotów obrabianych, opisanych nie stałymi wymiarami lecz parametrami, pozwalających na ich wykorzystanie w wielu programach sterujących (tworzenie własnych cykli obróbkowych).

8.3.1. Przykład dla obróbki frezarskiej

Zaprogramować obróbkę konturu krzywki (Rys. 143), której fragment ma postać spirali logarytmicznej, opisanej danymi parametrami. Przyjąć powierzchnię materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 10 mm. Rozpocząć od punktu (-41,0) w kierunku przeciwnym do ruchu wskazówek zegara.



Rys. 143. Szkic przedmiotu do przykładu obróbki z wykorzystaniem programowania strukturalnego

Rozwiązanie:

Równanie spirali logarytmicznej we współrzędnych biegunowych ma następującą postać:

$$R = A \cdot e^{\lambda \cdot \varphi} \tag{17}$$

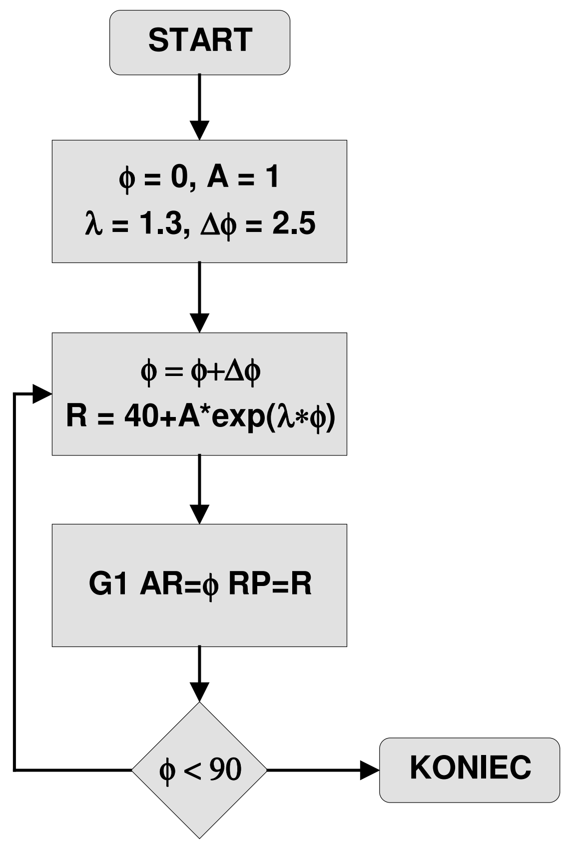
gdzie: A, λ – wielkości stałe;

R, φ [rad] – współrzędne biegunowe (promień i kąt).

W przedstawionej krzywce występują dwa symetryczne odcinki spirali logarytmicznej o równaniu:

$$R = 40 + 1 \cdot e^{1.3 \cdot \varphi} \tag{18}$$

Aby wykonać ruch po krzywej danej powyższym wzorem (zakładając brak odpowiedniej interpolacji w układzie sterowania) należy interpolować go odcinkami linii prostej o małej długości, co z wystarczającą dokładnością pozwoli odwzorować dowolny kształt. Algorytm obliczeń toru ruchu narzędzia w postaci schematu blokowego pokazano na Rys. 144. Występujący na nim parametr $\Delta\phi$ jest przyrostem kąta ϕ (w programowaniu biegunowym). Kąt ten dla pierwszej części odcinka spirali zmienia się od wartości początkowej 0° do 90°, stąd warunek przerwania pętli. Wartość o jaką zmienia się kąt zależy od dokładności, z jaką chcemy odwzorować kształt spirali – im mniejsza wartość tym większa dokładność. Drogą eksperymentów trzeba dobrać jej wartość poprawną (nie za dużą, ale też i nie za małą). Drugą część spirali można uzyskać przy użyciu analogicznego algorytmu z tą różnicą, że wartość kąta we wzorze na promień *R* musi być równa (180-ϕ).



Rys. 144. Schemat blokowy obliczeń toru ruchu dla odcinka spirali logarytmicznej

Program sterujący:

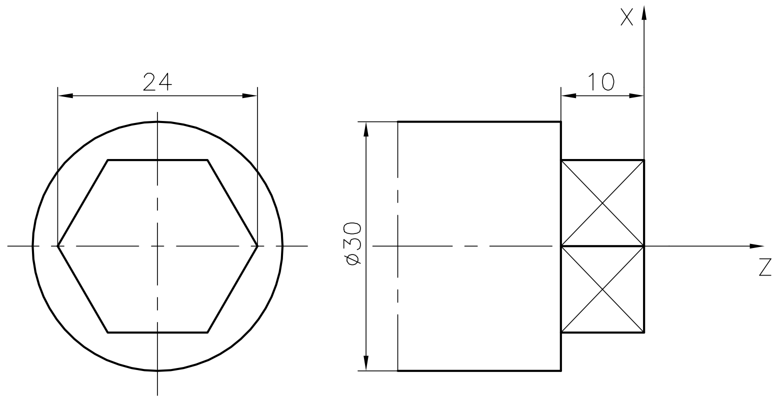
```
%_N_EX10_MPF
; 28-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S800 F100 M3 M8 M6
N15 G0 X-55 Y10
N20 Z3
N25 G1 Z-5
N30 G42 X-41 Y0
N35 G3 X41 CR=41
N40 MSG("RUCH PO SPIRALI")
N45 G111 X0 Y0 ;BIEGUN
; DANE WEJSCIOWE
N50 R1=0 ;FI
N55 R2=1 ;A
N60 R3=1.3 ;LAMBDA
N65 R4=0 ;R
N70 R5=2.5 ;DELTA_FI
; START 1 PETLI
PETLA1:
N80 R1=R1+R5                    ;NOWY KAT
N85 R6=R1*3.1415/180            ;RADIANY
N90 R4=40+(R2*EXP(R3*R6))       ;NOWY PROMIEN
N95 G1 RP=R4 AP=R1              ;RUCH
N100 IF R1<90 GOTOB PETLA1
; START 2 PETLI
N105 R1=90
PETLA2:
N115 R1=R1+R5                   ;NOWY KAT
N120 R6=(180-R1)*3.1415/180     ;RADIANY
N125 R4=40+(R2*EXP(R3*R6))      ;NOWY PROMIEN
N130 G1 RP=R4 AP=R1             ;RUCH
N135 IF R1<180 GOTOB PETLA2
N140 G0 G40 X-60
N145 Z100
N150 MSG("")
N155 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N160 M30
```

8.3.2. Przykład dla obróbki tokarskiej

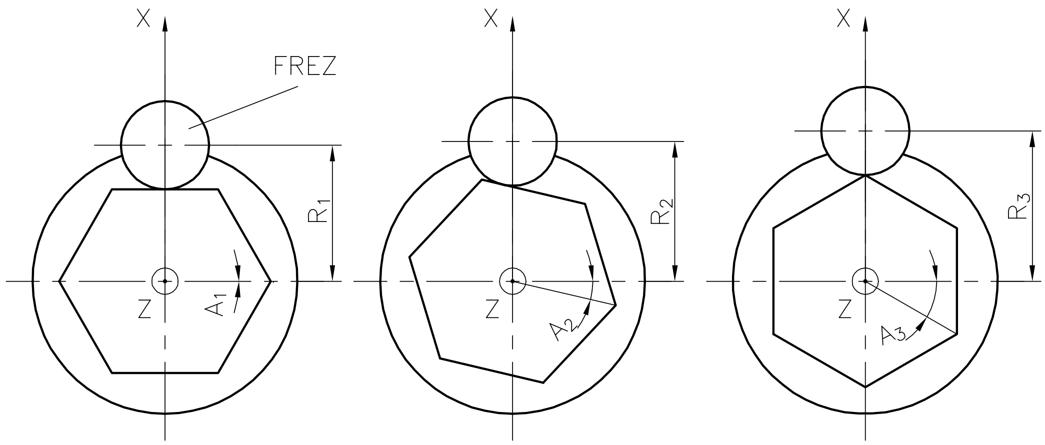
Zapisać program obróbki czopa o kształcie sześciokąta (Rys. 145) za pomocą freza palcowego o średnicy 10 mm (w głowicy z napędem narzędzi) na centrum tokarskim, wykorzystując funkcję pozycjonowania kątowego wrzeciona.

Rozwiązanie:

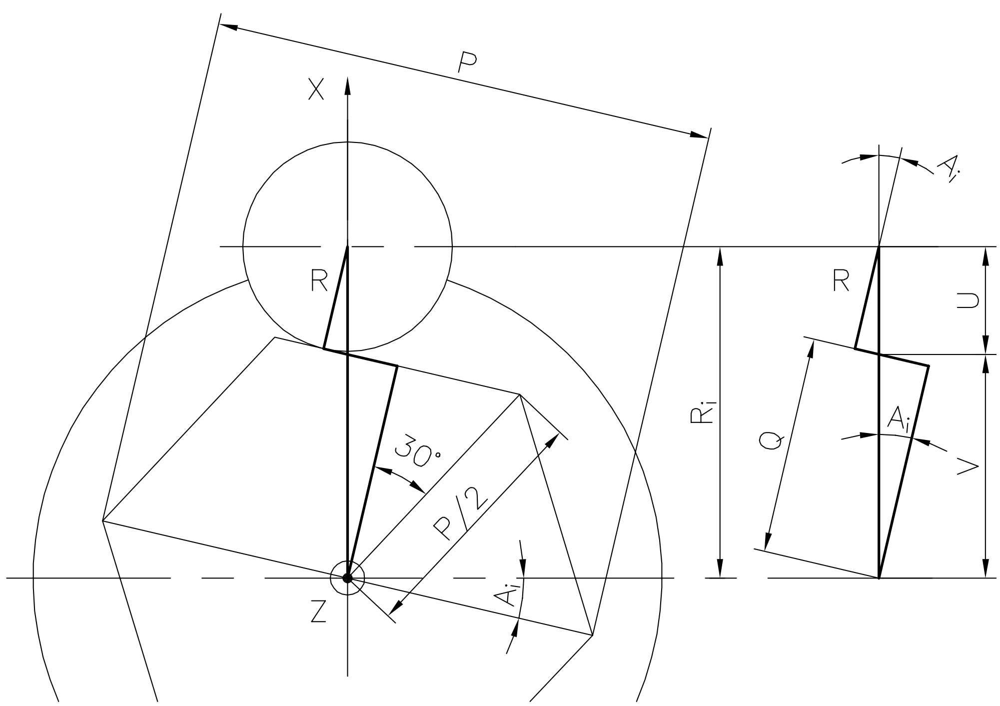
Obróbka polega na odpowiednim skojarzeniu położenia kątowego wrzeciona ze zmianami położenia liniowego narzędzia – Rys. 146. Należy zatem wyznaczyć zależność pomiędzy współrzędną R_i a kątem obrotu A_i w zakresie pełnego obrotu przedmiotu (0÷360°). Ponieważ przedmiot jest symetryczny wystarczy określenie tego związku w przedziale 0÷30°. Wielkości potrzebne do wyznaczenia wspomnianej zależności przedstawiono na Rys. 147.



Rys. 145. Szkic przedmi.powierzchni nieobrotowych)



Rys. 146. Kolejne fazy obróbki przy skojarzeniu położenia kątowego wrzeciona z położeniem liniowym freza w osi X



Rys. 147. Zasada wyznaczania zależności pomiędzy kątem A_i a promieniem R_i

Z Rys. 147 wynikają następujące zależności:

$$
\begin{aligned}
&Q = \frac{P \cos 30^\circ}{2} \\
&U = \frac{R}{\cos A_i}; \quad A_i \in \left\langle 0{,}30^\circ \right\rangle \\
&V = \frac{Q}{\cos A_i}; \quad A_i \in \left\langle 0{,}30^\circ \right\rangle \\
&R_i = U + V
\end{aligned}
\tag{19}
$$

Program sterujący:

```
%_N_EX14_MPF
; 13-09-2003
N5 G40 G54 G71 G90 G95 DIAMON
N10 MSG("FREZOWANIE CZOPA")
N15 T1 D1
; DANE WEJSCIOWE
N20 R1=5        ;PROMIEN FREZA R
N25 R2=24       ;P
N30 R3=0        ;Ai STARTOWE
N35 R4=0        ;Ri STARTOWE
N40 R5=1        ;SKOK Ai
; POCZATEK OBROBKI
N45 R100=2*R1+30+5  ;POZYCJA STARTOWA FREZA
N50 G0 X=R100
N55 Z-10
N60 SPOS=0
N65 G94 F50     ;PARAMETRY OBROBKI
N70 M33 H1000   ;START OBROTOW FREZA
N75 DIAMOF
; OBLICZENIA W PETLI
PETLA:
N85 R6=R3       ;ZMIENNA POMOCNICZA (KAT)
N90 IF R6>60 R6=R6 MOD 60
N95 IF R6>30 R6=60-R6
N100 R7=R2*COS(30)/2 ;Q
N105 R8=R1/COS(R6)   ;U
N110 R9=R7/COS(R6)   ;V
N115 R4=R8+R9        ;NOWE Ri
N120 G1 X=R4 SPOS=R3
N125 R3=R3+R5        ;NOWE Ai
N130 IF R3<360 GOTOB PETLA
N135 DIAMON
N140 MSG("")
N145 G53 T0 D0 G0 Z300 X300 M9 M35
N150 M30
```

Od bloku N20 przypisuje się R-parametrom zadane wymiary (np. promień freza). W bloku N45 obliczona zostaje pozycja wyjściowa narzędzia (zależna od jego promienia), a następnie włączony zostaje tryb programowania G94 (blok N65), charakterystyczny dla frezarek, uniezależniający posuw od obrotów wrzeciona (to

zostaje ustawione w trybie pozycjonowania kątowego – N60); jednocześnie ustawiony zostaje posuw minutowy F50. W bloku N70 zostają włączone prawe obroty narzędzia T1 (M33) o wartości 1000 obr/min (H1000). Jest to sposób przykładowy, pochodzący z konkretnej obrabiarki, dla innych ta funkcja może wyglądać inaczej. Po wyłączeniu programowania średnicowego (N75) rozpoczyna się pętla, w której wykonywane są obliczenia wartości promienia R_i przy zadanej wartości kąta A_i (N85÷N115). Początkowa wartość kąta A_i została ustalona jako 0° (N30), a w każdej kolejnej pętli jest ona zwiększana o zadany w bloku N40 przyrost (N125). Pętla jest wykonywana do czasu przekroczenia wartości kąta równej 360°. Właściwy ruch narzędzia jest wykonywany w bloku N120 – ustawienie nowej pozycji kątowej z jednoczesnym przemieszczeniem freza w osi X. Występująca w blokach N85÷N95 zmienna pomocnicza R6 ma za zadanie sprowadzić obliczenia do przyjętego wcześniej przedziału wartości kąta A_i 0÷30°.

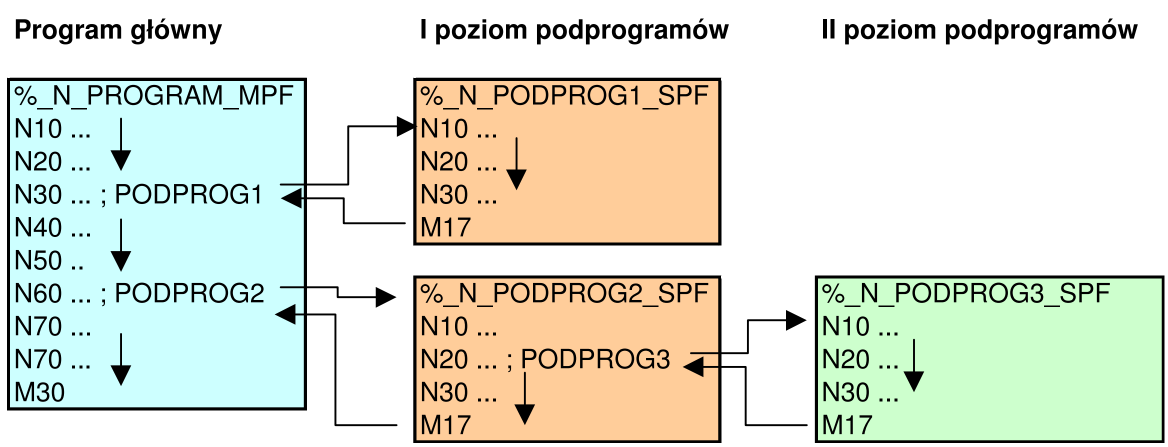
Po wykonaniu żądanej liczby powtórzeń pętli z powrotem ustawiane jest programowanie średnicowe (N135) oraz wykonywne jest wycofanie narzędzia do pozycji bezpiecznej (N145) z wyłączeniem jego obrotów (M35).

9. PODPROGRAMY

9.1. Wprowadzenie

Podprogramy to już wyższy poziom programowania. Są one bardzo podobne do funkcji czy procedur stosowanych w klasycznych językach programowania (Pascal, C). Ich zadaniem jest uczynienie programu bardziej przejrzystym, łatwiejszym w analizie, a często także ograniczenie jego objętości (dzięki wielokrotnemu wywoływaniu z programu głównego zamiast powtarzaniu w programie głównym identycznych bloków). Jeżeli jeszcze wprowadzić opcje parametryzacji i funkcje strukturalne (rozdz. 3) to podprogramy doprowadzą nas do powszechnie stosowanych cykli obróbkowych (rozdz. 7). Poznanie techniki podprogramów jest zatem pierwszym krokiem do zrozumienia cykli obróbkowych.

Czym zatem jest podprogram. Jego struktura w zasadzie jest identyczna jak programu głównego, różny jest tylko nagłówek i zakończenie. Podprogram może zawierać wszystkie adresy, jakie występują w programie głównym, może również zawierać wywołania innych podprogramów. Przy analizie i wykonaniu programu z wywołanymi podprogramami dla układu sterowania nie ma znaczenia, czy aktualny blok zapisany jest w programie głównym czy podprogramie – dla niego jest to jeden, nieprzerwany ciąg bloków, tylko zapisanych w różnych plikach. Schematycznie funkcjonowanie programu z wykorzystaniem podprogramów przedstawiono na Rys. 148. Dla różnych układów sterowania różny jest maksymalny poziom zagnieżdżenia podprogramu, w przypadku układu Sinumerik 840D wynosi on 11.



Rys. 148. Analiza i wykonanie programu sterującego z wywołaniami podprogramów

9.2. Wywoływanie podprogramów

Istnieje szereg różnych metod wywoływania podprogramów. Są one związane przede wszystkim ze sposobem przekazywania pewnych informacji z poziomu nadrzędnego, jak również z uwarunkowaniami historycznymi. Poniżej scharakteryzowano najważniejsze i najczęściej stosowane metody wywoływania podprogramów:

➔ Przez adres **L** – jest to metoda najdłużej funkcjonująca w Sinumeriku; warunkiem podstawowym jest ograniczenie nazewnictwa podprogramów – nazwa

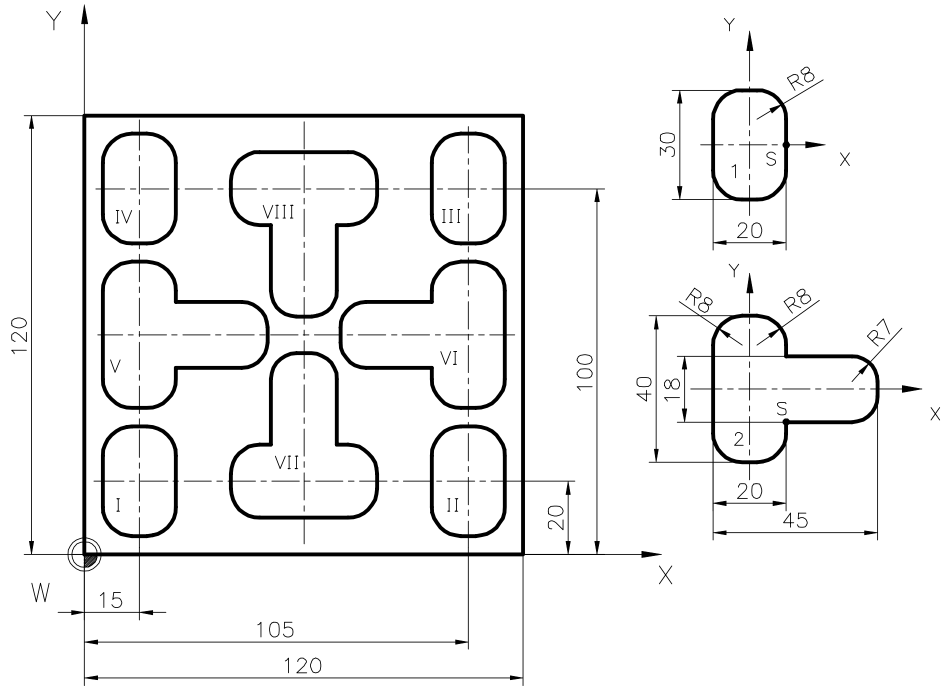
podprogramu musi być liczbą całkowitą z przedziału 1÷9999 (nazwa pliku 1.SPF÷9999.SPF). Wtedy odwołując się do podprogramu pod wartością adresu L podajemy numer tegoż podprogramu, np. wywołując podprogram 13 (plik 13.SPF) użyjemy słowa L13. Z adresem L często łączy się adres **P**. Jego wartość programuje liczbę wywołań danego podprogramu z tego samego punktu programu głównego (lub innego podprogramu). Jeżeli krotność ta wynosi 1 (najczęściej) to adres P można pominąć. Przykład wykorzystania podprogramu wywoływanego przez adres L podano w rozdz. 1.4.1.

➔ Przez nazwę podprogramu – w układzie sterowania Sinumerik 840D nazwy podprogramów nie muszą już ograniczać się do numeru z przedziału 1÷9999. Obecnie nazwa może zawierać również litery i inne znaki alfanumeryczne (ograniczenia dotyczące nazw podane są w dokumentacji firmowej). Wywołanie tak nazwanego podprogramu może się odbyć przez podanie jego nazwy. Dla przykładu, jeżeli podany w rozdz. 1.4.1 podprogram miał nazwę „_N_KONTUR_SPF" to program go wywołujący ma wtedy następującą postać:

```
%_N_EX05_MPF
; 18-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S1000 F100 M3 M8 M6
; ELEMENT 1
N15 TRANS X20 Y10
N20 KONTUR
; ELEMENT 2
N25 TRANS X60 Y10
N30 ASCALE X1.2 Y1.2
N30 KONTUR
; ELEMENT 3
N35 TRANS X20 Y60
N40 AROT RPL=20
N45 KONTUR
; ELEMENT 4
N50 TRANS X90 Y60
N55 AMIRROR X0
N60 KONTUR
N65 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N70 M30
```

9.2.1. Przykład

Zaprogramować ruch narzędzia po ośmiu elementach konturu (obróbka wewnętrzna) przedstawionych na Rys. 149 w przyjętym układzie współrzędnych przedmiotu (WKS) o początku w punkcie **W**. Przyjąć poziom materiału Z=0, głębokość obróbki Z=-5. Obróbkę wykonać frezem palcowym o średnicy 6 mm. Wykorzystać podprogramy obróbki dla podanych kształtów kieszeni w lokalnych układach współrzędnych, pokazanych na rysunku. Obróbkę rozpocząć od punktów **S** w kierunku przeciwnym do ruchu wskazówek zegara.



Rys. 149. Szkic przedmiotu do przykładu obróbki z wykorzystaniem podprogramu

Rozwiązanie:

Podprogram obróbki 1. elementu konturu:

```
%_N_KONTUR1_SPF
; 25-08-2003
N5 G0 X0 Y0
N10 Z3
N15 G1 Z-5
N20 G41 G247 X10 DISR=5
N25 Y15 RNDM=8
N30 X-10
N35 Y-15
N40 X10
N45 Y0 RNDM=0
N50 G40 G248 X0 DISR=5
N55 G0 Z50
N60 M17
```

Podprogram obróbki 2. elementu konturu:

```
%_N_KONTUR2_SPF
; 25-08-2003
N5 G0 X0 Y0
N10 Z3
N15 G1 Z-5
N20 G41 G147 X10 Y-9 DISR=1
N25 X35 RNDM=7
N30 Y9
N35 X10 RNDM=0
N40 Y20 RNDM=8
N45 X-10
N50 Y-20
N55 X10
N60 Y-9 RNDM=0
```

```
N65 G40 G148 X0 Y0 DISR=1
N70 G0 Z50
N75 M17
```

Program główny:

```
%_N_EX08_MPF
; 25-08-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S1000 F80 M3 M8 M6
N15 TRANS X15 Y20
N20 KONTUR1 ; OBR. I
N25 TRANS X105 Y20
N30 KONTUR1 ; OBR. II
N35 TRANS X105 Y100
N40 KONTUR1 ; OBR. III
N45 TRANS X15 Y100
N50 KONTUR1 ; OBR. IV
N55 TRANS X15 Y60
N60 KONTUR2 ; OBR. V
N65 ATRANS X90
N70 AMIRROR X0
N75 KONTUR2 ; OBR. VI
N80 TRANS X60 Y20
N85 AROT RPL=90
N90 KONTUR2 ; OBR. VII
N95 ATRANS X80
N100 AMIRROR X0
N105 KONTUR2 ; OBR. VIII
N110 TRANS
N115 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N120 M30
```

9.3. Zaawansowane metody wywoływania podprogramów

(W przygotowaniu)

10. PROGRAMOWANIE CYKLI OBRÓBKOWYCH

10.1. Informacje podstawowe

Cykle obróbkowe stanowią istotny składnik programów sterujących, pozwalający na znaczną automatyzację programowania, ograniczenie rozmiarów programów oraz umożliwiający szybką i prostą zmianę parametrów zabiegów, przez nie opisywanych. Pod pojęciem cykli obróbkowych rozumie się stałe, sparametryzowane podprogramy, umieszczone w układzie sterowania, służące do programowania obróbki typowych zabiegów, np. wiercenia, gwintowania, toczenia itp. Często (jak ma to miejsce w układzie sterowania Sinumerik) definiowanie cykli obróbkowych jest wspomagane przez moduł programowania dialogowego, pozwalający na graficzną prezentację znaczenia parametrów cyklu, ułatwiające poprawne nadanie im wartości w konkretnej sytuacji. Z uwagi na charakter obróbki cykle obróbkowe zostały podzielone na trzy grupy:

- ➔ cykle wiercenia;
- ➔ cykle frezowania;
- ➔ cykle toczenia.

Każda z powyższych grup zostanie w dalszych rozdziałach bardziej szczegółowo omówiona.

Przy wywoływaniu cykli należy zdefiniować ich parametry. Mogą one mieć różną postać w różnych układach sterowania, np. poprzez wykorzystanie R-parametrów. W układzie Sinumerik 810D/840D cykle obróbkowe są zdefiniowane w postaci procedur, których parametry są podawane razem z nazwą cyklu, np.

CYCLE100(34,67,2,90)

Gdzie CYCLE100 jest przykładową nazwą cyklu, natomiast w nawiasach podano wartości kolejnych parametrów tego cyklu. Mogą one być podane zarówno jako wartości stałe (konkretne liczby), jak i jako zmienne (np. R-parametry). Bliższe szczegóły dotyczące tego zagadnienia zawarte są w instrukcji układu sterowania. Przy podawaniu wartości parametrów cyklu może wystąpić sytuacja, kiedy możliwe jest opuszczenie któregoś parametru (np. skok gwintu może być zdefiniowany przez dwa parametry, w wywołaniu cyklu gwintowania podajemy tylko jeden). Wtedy zaznacza się ten fakt przez pozostawienie pustej pozycji w liście parametrów cyklu, np.:

CYCLE100(34,,2,90)

gdzie opuszczono drugi z parametrów (domyślnie układ sterowania przypisuje mu wartość zerową). Jeżeli pomijane wartości parametrów znajdują się na końcu listy parametrów można opuścić je bez pozostawienia pustej pozycji, np.

CYCLE100(34,67,2)

gdzie opuszczono ostatni, czwarty parametr. Należy mieć na uwadze, że nie dla wszystkich parametrów można nie definiować ich wartości. Należy również zwracać uwagę na sposób interpretacji wartości parametrów, np. dla niektórych istotna jest nie tylko wartość ale i znak tej wartości.

10.2. Wywoływanie cykli obróbkowych

Wywołanie cykli obróbkowych (podobnie jak podprogramów – rozdz. 6) może mieć dwojaki charakter:

- ➔ niemodalny (tylko w jednym bloku);
- ➔ modalny (w każdym bloku z ruchem narzędzia aż do odwołania).

Wywołanie niemodalne (najczęściej stosowane) polega na umieszczeniu w bloku, w którym chcemy wykonać cykl obróbkowych, adresu z nazwą cyklu i listą wartości parametrów (tak, jak przedstawiono powyżej).

Wywołanie modalne polega na umieszczeniu w bloku oprócz wywołania samego cyklu również adresu MCALL (rozdz. 6.3), dzięki czemu w następnych blokach z instrukcjami ruchu narzędzia, po wykonaniu tych ruchów, automatycznie wywoływany jest również cykl modalny. Odwołanie wykonywania cyklu modalnego następuje po zaprogramowaniu adresu MCALL bez wywołania cyklu (wywołanie z cyklem powoduje zmianę cyklu modalnego), np.

```
N10 MCALL CYCLE100(34,67,2)      ; modalne przywołanie cyklu
N15 G0 X20 Y30                   ; pierwsze uruchomienie cyklu
N20 Y50                          ; drugie uruchomienie cyklu
N25 Y100                         ; trzecie uruchomienie cyklu
N30 MCALL                        ; odwołanie cyklu modalnego
```

10.3. Cykle wiercenia

Cykle wiercenia obejmują zabiegi obróbki otworów (przy użyciu różnych narzędzi: wierteł, pogłębiaczy, rozwiertaków, gwintowników itp.). Jest to jedyna grupa cykli, która została znormalizowana (dokładniej zaś cykle 81 do 89).

Cykle wiercenia generalnie działają wg następującego schematu:

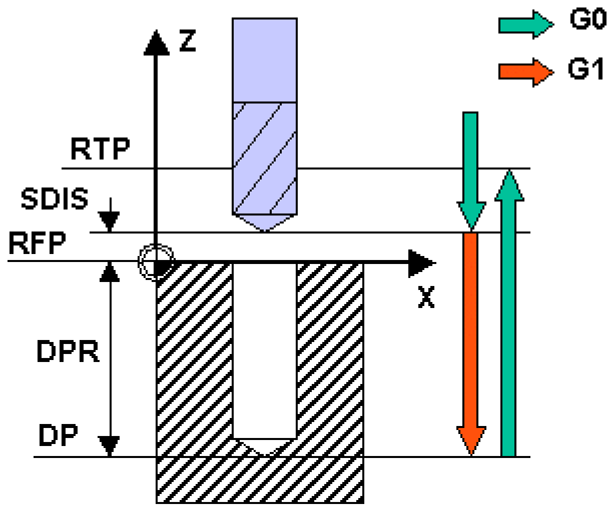
1. Dosunięcie ruchem szybkim narzędzia w osi Z do płaszczyzny bezpiecznej (tj. w bezpiecznej odległości od materiału);
2. Obróbka ruchem roboczym do zadanej głębokości;
3. Wycofanie ruchem szybkim narzędzia do płaszczyzny wycofania (tj. płaszczyzny na której może się odbywać ruch narzędzia bez ryzyka kolizji) bez wyłączania obrotów wrzeciona.

Wśród parametrów cykli występuje jeszcze płaszczyzna odniesienia – jest to płaszczyzna, od której teoretycznie narzędzie powinno rozpocząć obróbkę, jednak z uwagi na bezpieczny dojazd do materiału obrabianego faktycznie ten ruch rozpoczyna się od płaszczyzny bezpiecznej. Poszczególne cykle różnią się sposobem realizacji podanego schematu co szczegółowo omówiono poniżej, przedstawiając nazwę cyklu, listę jego parametrów, ich typ i znaczenie oraz krótki opis działania cyklu. Parametry, które mogą zostać pominięte w wywołaniu cyklu oznaczono gwiazdką (*). Na rysunkach schematycznie pokazano sekwencję czynności wykonywanych w ramach cyklu.

10.3.1. Wiercenie, nawiercanie – CYCLE81

CYCLE81(RTP, RFP, SDIS, DP, DPR) (Rys. 150)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość wiercenia (absolutnie)
DPR*	Real	Ostateczna głębokość wiercenia (przyrostowo, bez znaku)



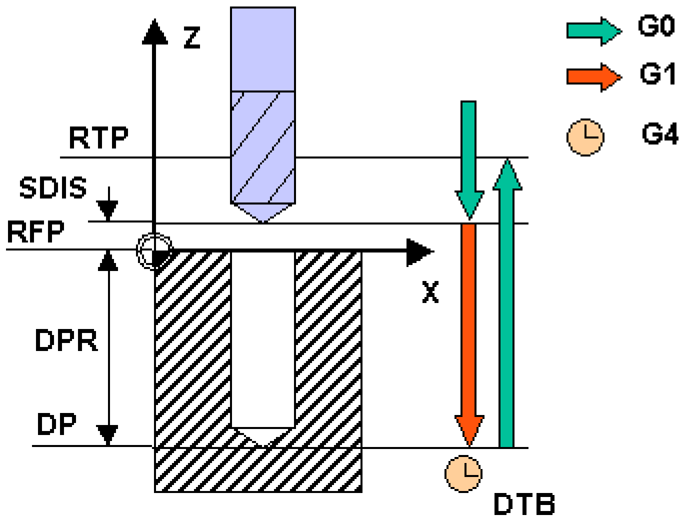
Rys. 150. Cykl wiercenia CYCLE81

Cykl CYCLE81 jest najprostszym wśród cykli wiertarskich. Stosowany jest przede wszystkim do wiercenia otworów o małej głębokości, wytaczania zgrubnego a także nawiercania (wykonywania nawiercenia przed właściwym zabiegiem wiercenia)

10.3.2. Wiercenie, pogłębianie – CYCLE82

CYCLE82(RTP, RFP, SDIS, DP, DPR, DTB) (Rys. 151)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość wiercenia (absolutnie)
DPR*	Real	Ostateczna głębokość wiercenia (przyrostowo, bez znaku)
DTB*	Real	Czas postoju na ostatecznej głębokości wiercenia



Rys. 151. Cykl wiercenia CYCLE82

W stosunku do cyklu CYCLE81 po osiągnięciu głębokości wiercenia może być zaprogramowany postój czasowy (DTB). Celem postoju może być złamanie wióra przy wierceniu lub obróbka dna otworu przy pogłębianiu pogłębiaczem walcowym czy stożkowym. Postój może być zaprogramowany w sekundach (DTB>0) lub obrotach wrzeciona (DTB<0) – podobnie jak w innych cyklach wiertarskich.

10.3.3. Wiercenie głębokiego otworu – CYCLE83

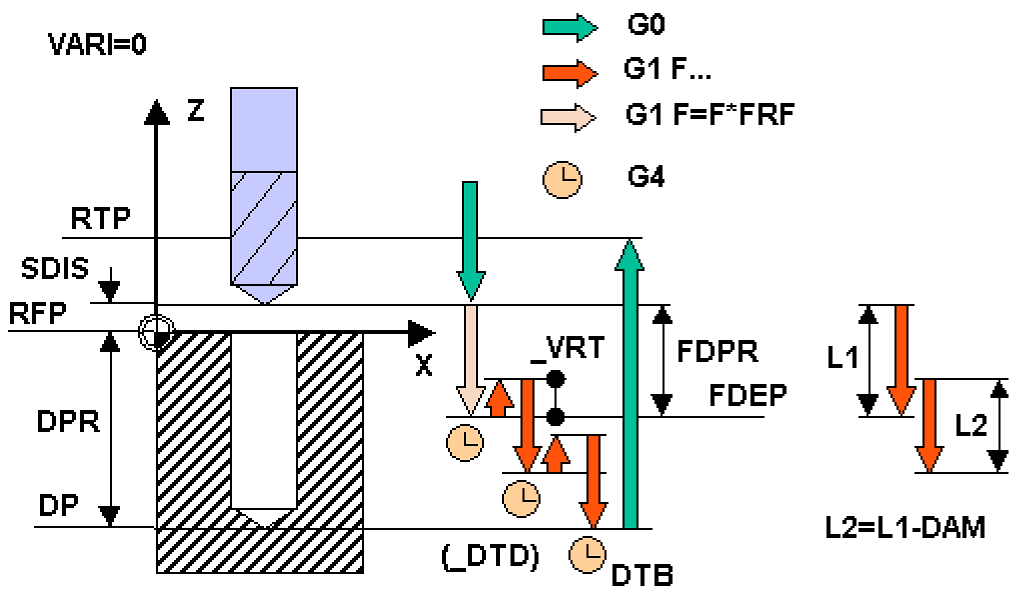
CYCLE83(RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI, _AXN, _MDEP, _VRT, _DTD, _DIS1) (Rys. 152, Rys. 153)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość wiercenia (absolutnie)
DPR*	Real	Ostateczna głębokość wiercenia (przyrostowo, bez znaku)
FDEP*	Real	Pierwsza głębokość wiercenia (absolutnie)
FDPR*	Real	Pierwsza głębokość wiercenia (przyrostowo, bez znaku)
DAM	Real	Degresja głębokości wiercenia (bez znaku)
DTB*	Real	Czas oczekiwania na głębokości wiercenia (łamanie wiórów)
DTS*	Real	Czas oczekiwania początkowym (odwiórowanie)
FRF*	Real	Współczynnik posuwu dla pierwszej głębokości wiercenia (0,001...1)
VARI	Integer	Rodzaj obróbki: Łamanie wiórów => 0, Usuwanie wiórów => 1
_AXN*	Integer	Oś narzędzia: 1, 2 lub 3 oś geometryczna (lub zadana funkcjami G17/G18/G19)
MDEP*	Real	Minimalna głębokość jednego wiercenia
_VRT*	Real	Droga wycofania przy łamaniu wiórów (VARI = 0)
_DTD*	Real	Czas postoju na ostatniej głębokości wiercenia
_DIS1*	Real	Odstęp od materiału przy wejściu po odwiórowaniu (VARI = 1): Zadany > 0 Obliczony automatycznie = 0

Cykl CYCLE83 jest przeznaczony do obróbki tzw. otworów głębokich, tj, takich, w których stosunek głębokości do średnicy przekracza wartość 1,5÷2. Wtedy narzędzie nie może od razu wiercić na pełną głębokość, konieczne jest kilkukrotne wejście narzędzia w materiał, przy czym możliwe są dwa rodzaje obróbki:

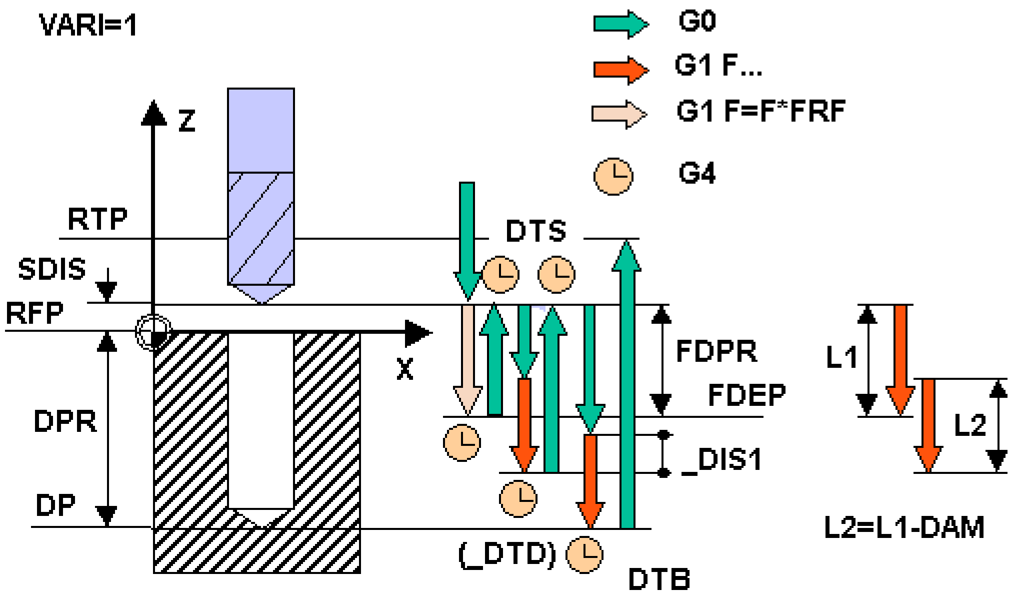
- ➔ z łamaniem wióra (Rys. 152) – po osiągnięciu częściowej głębokości następuje zatrzymanie posuwu przy włączonych obrotach wrzeciona (z ewentualnym wycofaniem narzędzia – _VRT); dzięki temu następuje złamanie wióra i łatwiejsze usunięcie go z wierconego otworu; po zadanym czasie postoju następuje wiercenie do kolejnej głębokości wiercenia.
- ➔ z usuwaniem wiórów (Rys. 153) – po osiągnięciu częściowej głębokości wiercenia następuje wycofanie narzędzia z otworu; po zadanym czasie postoju ruchem szybkim narzędzie powraca na poprzednią głębokość (pomniejszoną o odstęp bezpieczeństwa) i wierci do kolejnego poziomu obróbki.

Wybór jednej z powyższych strategii (jak również degresji głębokości wiercenia) zależy od głębokości otworu, jego średnicy, usytuowania (poziomy, pionowy), materiału obrabianego, sposobu chłodzenia (zewnętrzne, przez rdzeń wiertła) itp.



Rys. 152. Cykl wiercenia CYCLE83 z łamaniem wióra

Pierwsza głębokość wiercenia, w zależności od głębokości otworu, przyjmowana jest w granicach 1,5÷2,5*D* (*D* – średnica otworu). Przy większej wartości można zmniejszać posuw (FRF). Zmniejszanie posuwu może być również stosowane przy wierceniu otworów w odlewach z nieusuniętym naskórkiem (zwykle o większej twardości niż rdzeń).



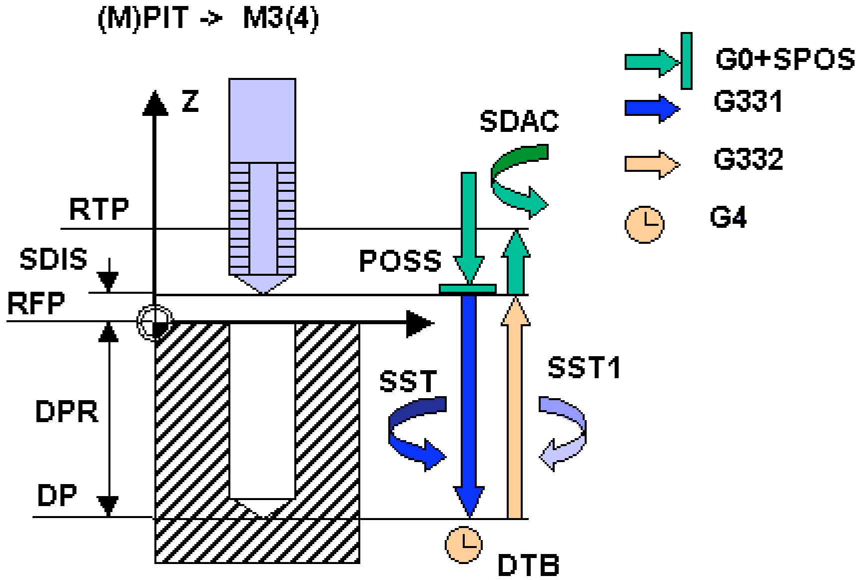
Rys. 153. Cykl wiercenia CYCLE83 z usuwaniem wiórów

10.3.4. Gwintowanie bez użycia uchwytu kompensacyjnego – CYCLE84

CYCLE84(RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1, _AXN) (Rys. 154)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB*	Real	Czas oczekiwania na głębokości ostatecznej (łamanie wiórów)
SDAC	Integer	Kierunek obrotów po zakończeniu cyklu : 3, 4 albo 5
MPIT*	Integer	Skok gwintu jako wielkość gwintu (z przedznakiem). Zakres wartości: 3 (dla M3) ... 48 (dla M48) Przedznak określa kierunek zwoju gwintu („+" prawy, „-„ lewy)
PIT*	Real	Skok gwintu jako wartość (z przedznakiem) Zakres wartości: 0,001 ... 2000,000 mm Przedznak określa kierunek zwoju gwintu („+" prawy, „-„ lewy)
POSS	Real	Pozycja wrzeciona dla zorientowanego zatrzymania (w stopniach)
SST	Real	Prędkość obrotowa dla gwintowania
SST1	Real	Prędkość obrotowa dla wycofania
_AXN*	Integer	Oś narzędzia: 1, 2 lub 3 oś geometryczna (lub zdefiniowana przez G17/G18/G19)

Uwaga!! Cykl CYCLE84 może być stosowany wtedy, gdy wrzeciono przewidziane do gwintowania jest technicznie w stanie przejść na pracę z regulacją położenia kątowego.



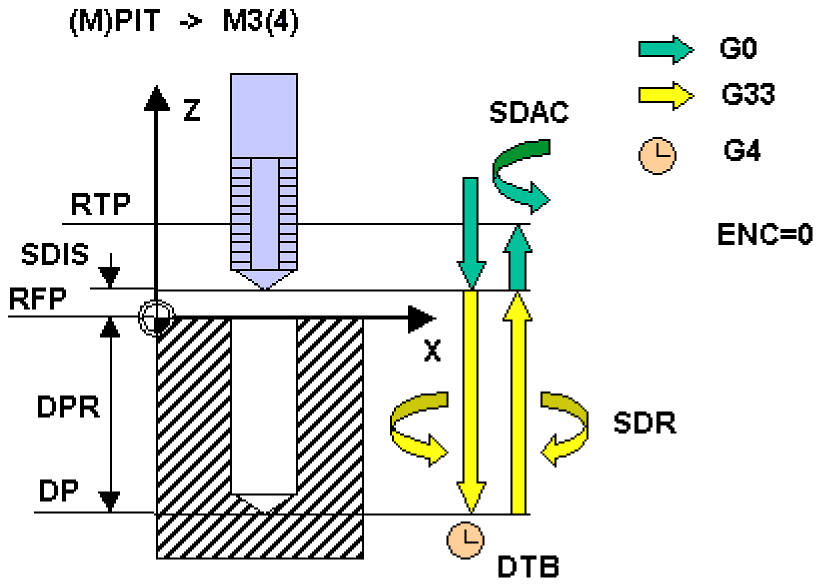
Rys. 154. Cykl gwintowania CYCLE84

Cykl CYCLE84 jest przeznaczony do nacinania gwintów gwintownikiem, przy czym nie jest wymagane stosowanie oprawki kompensacyjnej (z osiową kompensacją długości narzędzia). Wynika to z dokładnego sprzężenia ruchu obrotowego i posuwowego narzędzia (funkcje G331 i G332). Ten cykl wykorzystuje się jednak rzadziej niż CYCLE840, opisany w następnym rozdziale.

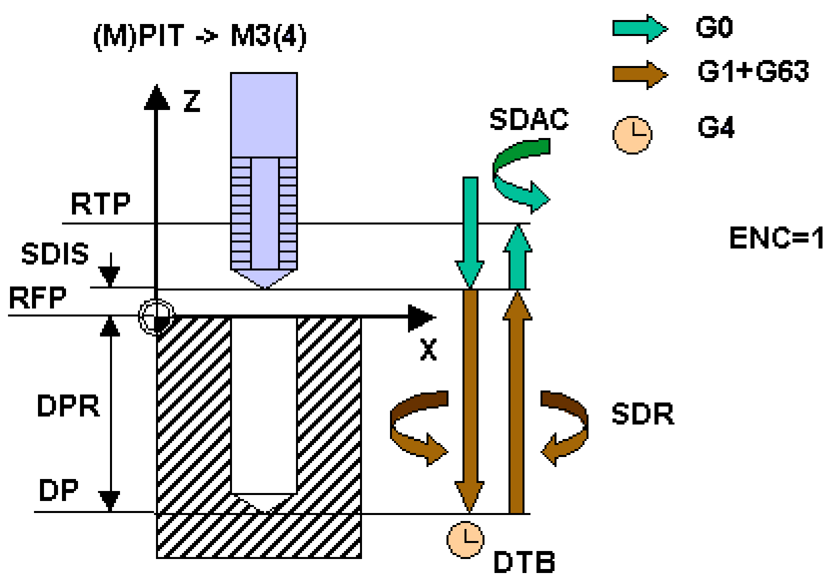
10.3.5. Gwintowanie z użyciem uchwytu kompensacyjnego – CYCLE840

CYCLE840(RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT)
(Rys. 155, Rys. 156)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB*	Real	Czas oczekiwania na pełnej głębokości gwintu
SDR	Integer	Kierunek obrotów dla wycofania: 0 – automatyczne odwrócenie kierunku obrotów; 3 – dla M3; 4 – dla M4;
SDAC	Integer	Kierunek obrotów po zakończeniu cyklu: 3 (M3), 4 (M4) lub 5 (M5)
ENC	Integer	Gwintowanie otworu: 0 – z koderem (Rys. 155); 1 – bez kodera (Rys. 156);
MPIT*	Real	Skok gwintu jako wielkość gwintu. Zakres wartości: 3 (M3)...48 (M48)
PIT*	Real	Skok gwintu jako wartość. Zakres wartości: 0,001 ... 2000,000 mm



Rys. 155. Cykl gwintowania CYCLE840 z koderem



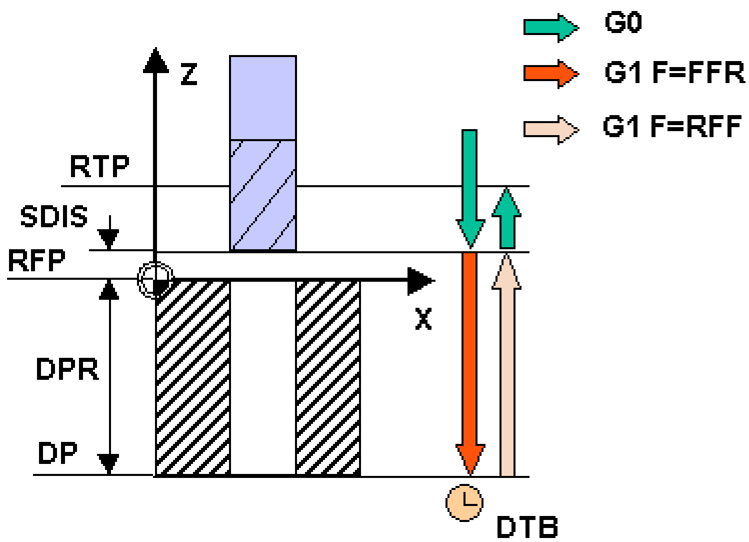
Rys. 156. Cykl gwintowania CYCLE840 bez kodera

Gwintowanie bez kodera (czyli tzw. „elektronicznej gitary", sprzęgającej ruch posuwowy w osiach liniowych z ruchem obrotowym wrzeciona), realizowane jest za pomocą interpolacji liniowej G1. Dodatkowo (co dokładnie opisano w rozdz. 3.3) programowana jest funkcja G63. Gwintowanie z koderem odbywa się z użyciem interpolacji spiralnej G33. Z uwagi na mniejszą dokładność sprzężenia ruchu obrotowego z ruchem posuwowym zaleca się mocować gwintownik w oprawce kompensacyjnej (osiowe zmiany długości całego narzędzia).

10.3.6. Rozwiercanie 1 – CYCLE85

CYCLE85(RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF) (Rys. 157)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB*	Real	Czas oczekiwania na głębokości rozwiercania (łamanie wiórów)
FFR	Real	Posuw roboczy
RFF	Real	Posuw przy wycofywaniu



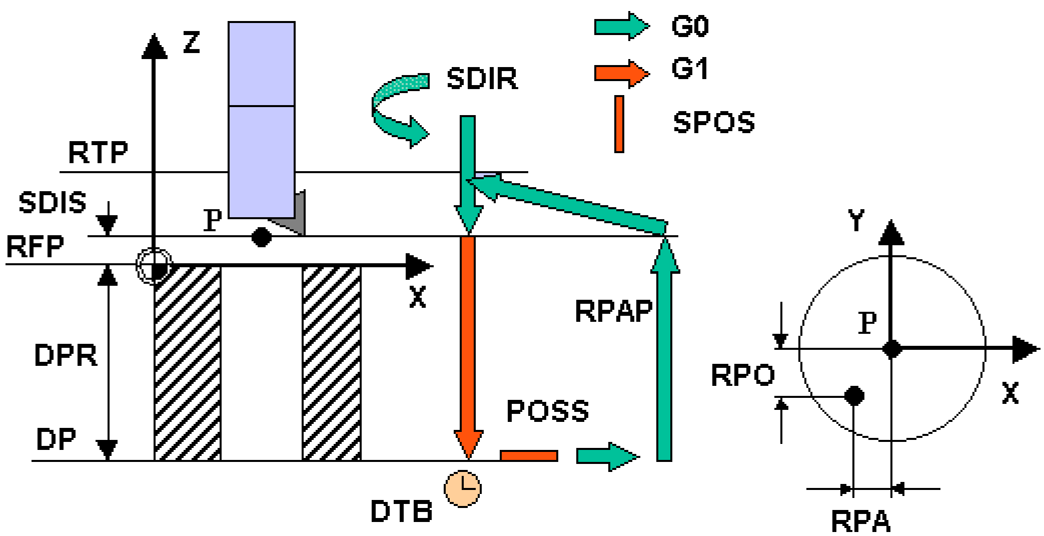
Rys. 157. Cykl rozwiercania 1 CYCLE85

Niniejszy cykl (oraz następne) mimo swej nazwy mogą być stosowane dla różnych zabiegów. Cykl CYCLE85 zgodnie ze swoją nazwą jest zalecany dla rozwiercania (szczególnie wykańczającego) oraz innych obróbek gładkościowych dzięki możliwości dokładnego wpływu na ślady po obróbce (sterowanie posuwem wgłębnym i przy wycofaniu).

10.3.7. Rozwiercanie 2 – CYCLE86

CYCLE86(RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS) (Rys. 158)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB*	Real	Czas oczekiwania na głębokości rozwiercania (łamanie wiórów)
SDIR	Integer	Kierunek obrotów przy obróbce: 3 (M3), 4 (M4)
RPA	Real	Droga wycofania w odciętej (przyrostowo, z przedznakiem)
RPO	Real	Droga wycofania w rzędnej (przyrostowo, z przedznakiem)
RPAP	Real	Droga wycofania w aplikacie (przyrostowo, z przedznakiem)
POSS	Real	Pozycja zorientowanego zatrzymania wrzeciona (w stopniach)



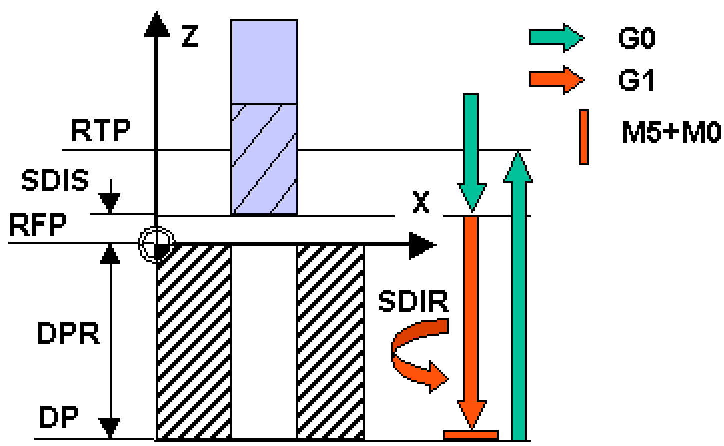
Rys. 158. Cykl rozwiercania 2 CYCLE86

CYCLE86 jest cyklem charakterystycznym dla obróbki narzędziami niesymetrycznymi (wytaczaki jednoostrzowe). W praktyce jest stosowany przede wszystkim przy wytaczaniu wykańczającym (przy wycofaniu narzędzie nie styka się z powierzchnią otworu). Programując odsunięcie narzędzie po zakończonej obróbce należy dokładnie uzgodnić położenie kątowe stopu pozycjonowanego wrzeciona (z uwagi na położenie ostrza), a także dopuszczalne wartości samych odsunięć (niebezpieczeństwo kolizji).

10.3.8. Rozwiercanie 3 – CYCLE87

CYCLE87(RTP, RFP, SDIS, DP, DPR, SDIR) (Rys. 159)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
SDIR	Integer	Kierunek obrotów w trakcie obróbki: 3 (M3), 4 (M4)



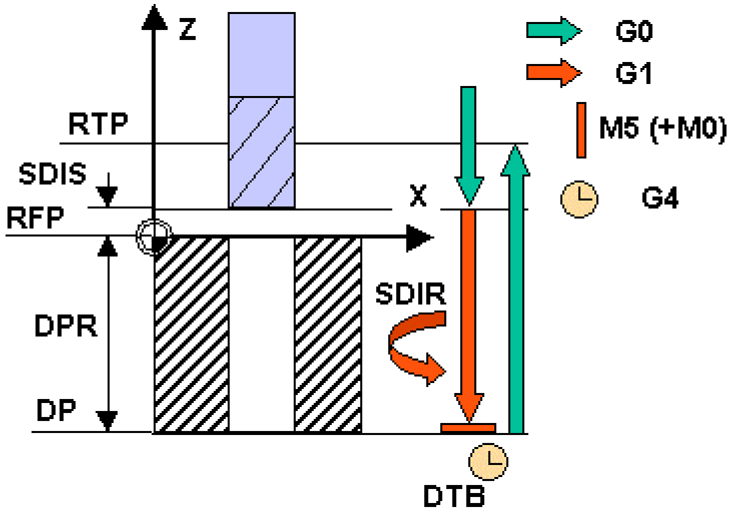
Rys. 159. Cykl rozwiercania 3 CYCLE87

W cyklu CYCLE87 po osiągnięciu pełnej głębokości obróbki następuje zatrzymanie obrotów wrzeciona (M5) oraz bezwarunkowe zatrzymanie wykonywania programu (M0). Kontynuacja wykonania programu po naciśnięciu przycisku NC-START. CYCLE87 wymaga zatem obecności operatora.

10.3.9. Rozwiercanie 4 – CYCLE88

CYCLE88(RTP, RFP, SDIS, DP, DPR, DTB, SDIR) (Rys. 160)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB	Real	Czas oczekiwania na ostatecznej głębokości rozwiercania
SDIR	Integer	Kierunek obrotów w trakcie obróbki: 3 (M3), 4 (M4)



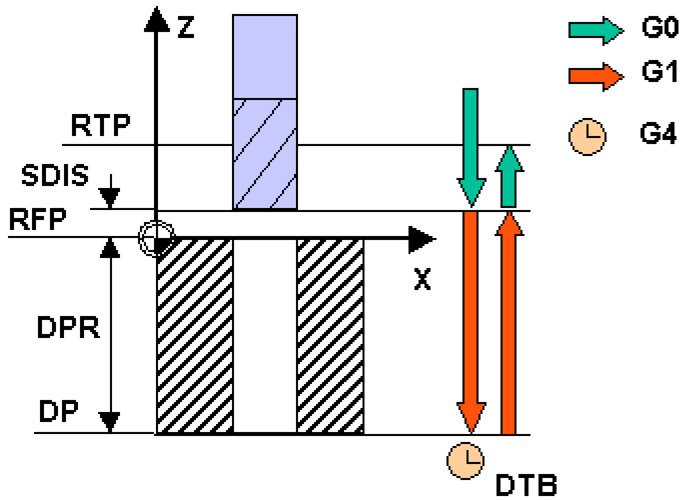
Rys. 160. Cykl rozwiercania 4 CYCLE88

CYCLE88 jest zbliżony w działaniu do CYCLE87. Po zakończeniu ruchu roboczego można zaprogramować postój czasowy (DTB). Dodatkowo, bezwarunkowe zatrzymanie programu (M0) nie musi zostać wykonane. Decyduje o tym wartość zmiennej nastawczej cykli _ZSD[5] (ang. *Cycles Setting Data*), ustawianej z poziomu pulpitu operatorskiego. Jeżeli wartość tej zmiennej jest równa 0 to aktywowana jest funkcja M5 i M0, w przeciwnym przypadku wykonywane jest tylko zatrzymanie obrotów.

10.3.10. Rozwiercanie 5 – CYCLE89

CYCLE89(RTP, RFP, SDIS, DP, DPR, DTB) (Rys. 161)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość gwintowania (absolutnie)
DPR*	Real	Ostateczna głębokość gwintowania (przyrostowo, bez znaku)
DTB*	Real	Czas oczekiwania na głębokości rozwiercania (łamanie wiórów)

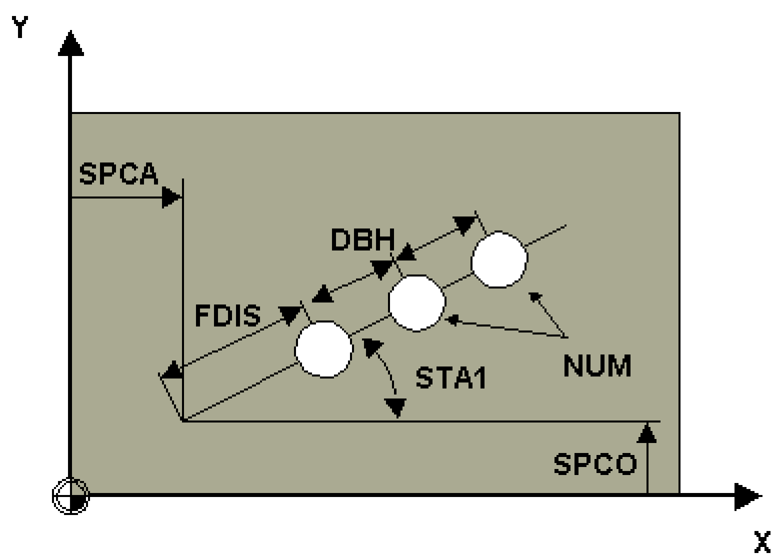


Rys. 161. Cykl rozwiercania 5 CYCLE89

10.3.11. Rząd otworów – HOLES1

HOLES1(SPCA, SPCO, STA1, FDIS, DBH, NUM) (Rys. 162)

SPCA	Real	Odcięta punktu odniesienia (absolutnie)
SPCO	Real	Rzędna punktu odniesienia (absolutnie)
STA1	Real	Kąt do odciętej, zakres wartości: -180° < STA1 ≤ 180°
FDIS	Real	Odstęp pierwszego otworu od punktu odniesienia (bez znaku)
DBH	Real	Odstęp między otworami (bez znaku)
NUM	Integer	Ilość otworów



Rys. 162. Rząd otworów HOLES1

Przy pomocy niniejszego cyklu (oraz dwóch kolejnych, opisanych w rozdz. 7.3.12 i 7.3.13) wykonuje się obróbkę otworów leżących w pewnym regularnym układzie – dla cyklu HOLES1 jest to układ rzędowy. Rodzaj obróbki otworów jest określany

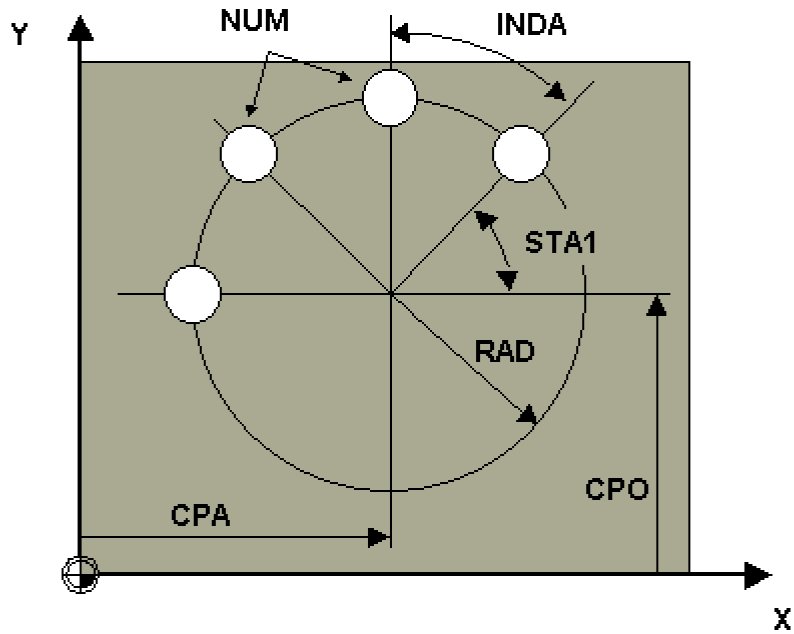
przez wybrany uprzednio modalnie cykl wiercenia (CYCLE81 ÷ CYCLE89). Właściwie można je zaliczyć do cykli frezarskich, gdyż tylko na takich obrabiarkach obróbka regularnych układów otworów ma sens.

Wewnętrznie w cyklu, dla uniknięcia zbędnych ruchów jałowych, następuje na podstawie rzeczywistej pozycji osi płaszczyzn i geometrii rozmieszczenia otworów wybór drogi narzędzia. Następnie następuje kolejne dosuwanie przesuwem szybkim do pozycji obróbki i realizacja modalnego cyklu wiercenia.

10.3.12. Kołowy układ otworów – HOLES2

HOLES2(CPA, CPO, RAD, STA1, INDA, NUM) (Rys. 163)

CPA	Real	Odcięta punktu środka koła (absolutnie)
CPO	Real	Rzędna punktu środka koła (absolutnie)
RAD	Real	Promień koła (bez znaku)
STA1	Real	Kąt początkowy, zakres wartości: $-180° \leq STA1 < 180°$
INDA	Real	Kąt pomiędzy otworami (ze znakiem)
NUM	Integer	Ilość otworów



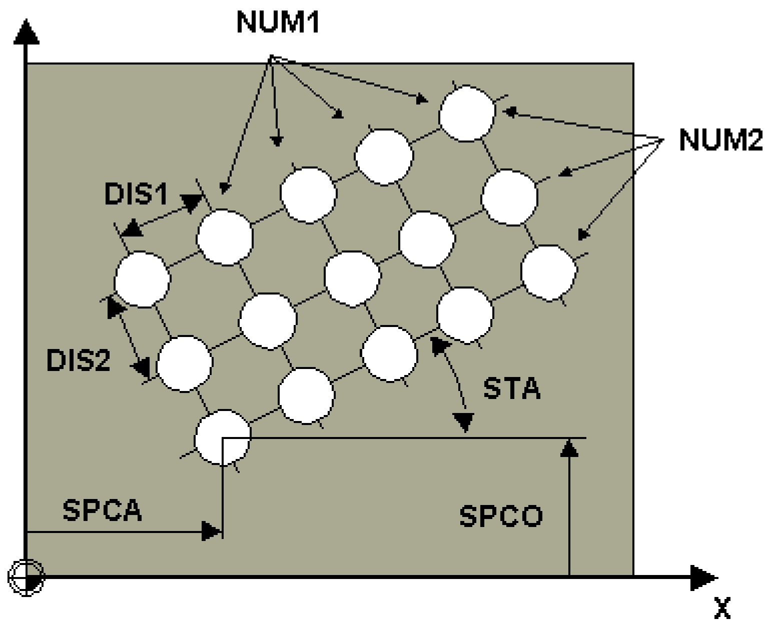
Rys. 163. Kołowy układ otworów HOLES2

Działanie cyklu podobne jak do opisanego powyżej, różnica polega na innym układzie otworów.

10.3.13. Macierz prostokątna otworów – CYCLE801

CYCLE801(SPCA, SPCO, STA, DIS1, DIS2, NUM1, NUM2) (Rys. 164)

SPCA	Real	Odcięta punktu bazowego (absolutnie)
SPCO	Real	Rzędna punktu bazowego (absolutnie)
STA	Real	Kąt do odciętej
DIS1	Real	Odległość między kolumnami (bez znaku)
DIS2	Real	Odległość między wierszami (bez znaku)
NUM1	Integer	Liczba kolumn
NUM2	Integer	Liczba wierszy



Rys. 164. Macierz prostokątna otworów CYCLE801

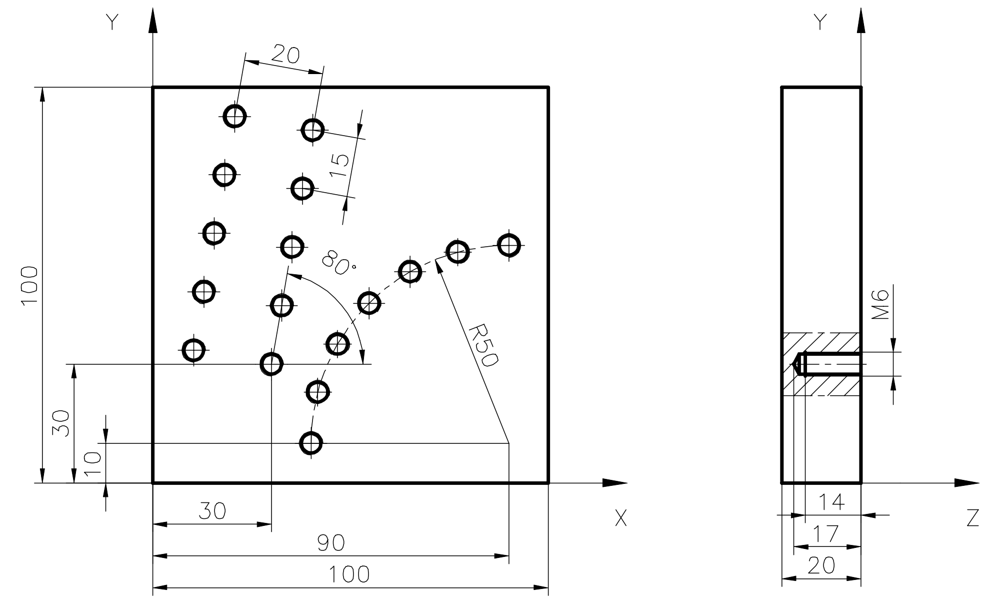
10.3.14. Przykład

Zaprojektować obróbkę otworów w płycie wg Rys. 165.

Rozwiązanie:

```
%_N_EX16_MPF
; 25-10-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 MSG("NAWIERCANIE")
N15 T1 D1 S1500 F250 M3 M8 M6
N20 G0 Z20
N25 MCALL CYCLE81(10,0,2,,3)
N30 HOLES2(90,10,50,90,15,7)
N35 CYCLE801(30,30,80,15,20,5,2)
N40 MCALL
N45 G0 X200 Y200 Z100
N50 MSG("WIERCENIE D4")
N55 T2 D1 S700 F200 M3 M8 M6
N60 MCALL CYCLE83(5,0,2,-17,,,5,1,-2,-3,0.8,1,,,,,)
N65 HOLES2(90,10,50,90,15,7)
```

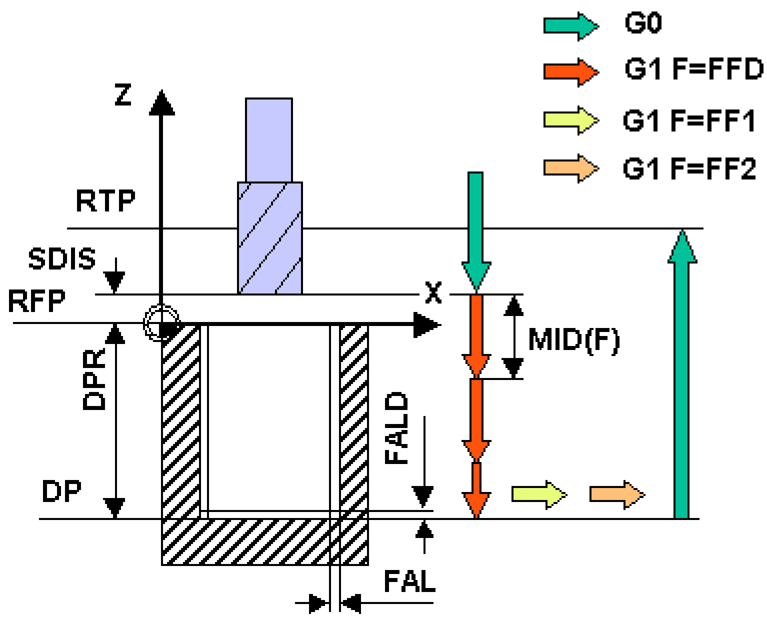
```
N70 CYCLE801(30,30,80,15,20,5,2)
N75 MCALL
N80 G0 X200 Y200 Z100
N85 MSG("GWINTOWANIE M5")
N90 T3 D1 S700 M8 M6
N95 MCALL CYCLE840(5,0,2,-14,,1,4,3,0,5,)
N100 HOLES2(90,10,50,90,15,7)
N105 CYCLE801(30,30,80,15,20,5,2)
N110 MCALL
N115 MSG()
N120 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N125 M30
```



Rys. 165. Szkic przedmiotu do przykładu obróbki z wykorzystaniem cykli wiertarskich

10.4. Cykle frezowania

Cykle frezowania stanowią zbiór charakterystycznych dla danego układu sterowania podprogramów obróbki powierzchni płaskich, rowków itp., wykonywanych na frezarkach. Na Rys. 166 przedstawiono wspólne dla wszystkich cykli parametry wzdłuż osi narzędzia (pozostałe zawarte są w opisach poszczególnych cykli). Część z nich występowała już w cyklach wiertarskich (RTP, RFP, SDIS, DP, DPR) i ma takie samo znaczenie – wymiarowanie obróbki w osi dosuwowej. Pozostałe są typowe dla obróbki frezarskiej – wymiarowanie naddatku na obróbkę wykańczającą (FAL, FALD), głębokość dosuwu zgrubnego i wykańczającego (MID, MIDF) czy posuw przy różnych rodzajach obróbki – zgrubnej, wykańczającej (FFD, FF1, FF2). W niniejszym skrypcie opisano tylko najczęściej stosowane cykle frezowania – pełny wykaz cykli i opis ich parametrów znajduje się w dokumentacji firmowej.



Rys. 166. Ogólne parametry cykli frezarskich

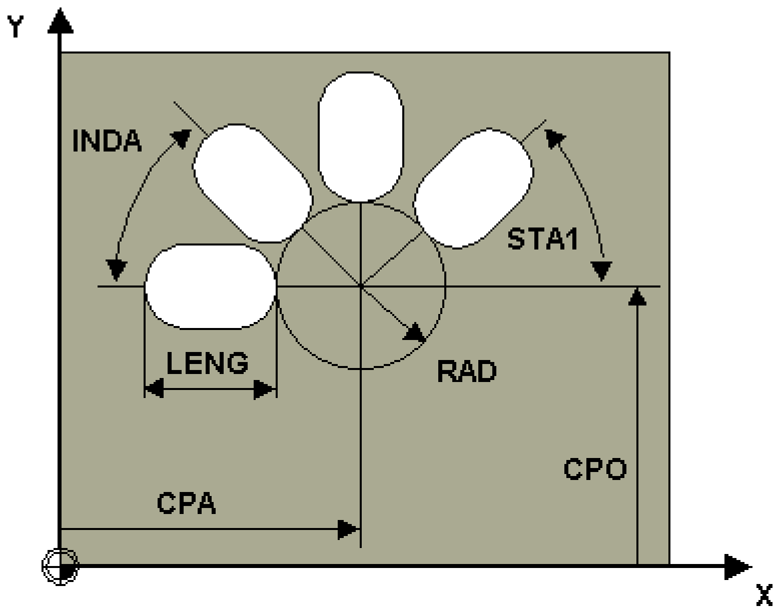
10.4.1. Rowki podłużne na okręgu – LONGHOLE

LONGHOLE(RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID) (Rys. 167)

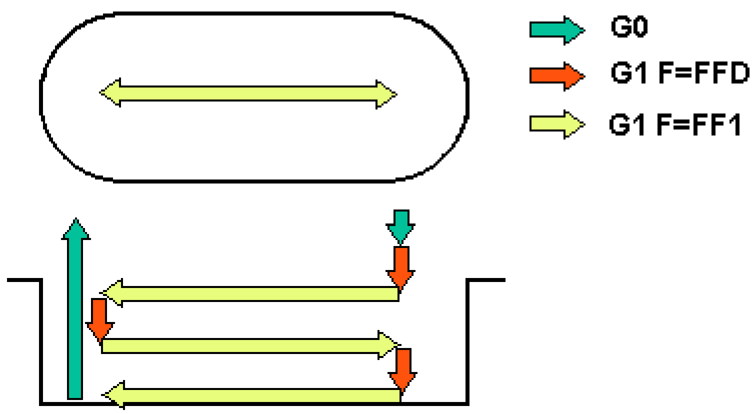
RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość rowka (absolutnie)
DPR*	Real	Ostateczna głębokość rowka (przyrostowo, bez znaku)
NUM	Integer	Ilość rowków
LENG	Real	Długość rowków (bez znaku)
CPA	Real	Odcięta punktu środkowego (absolutnie)
CPO	Real	Rzędna punktu środkowego (absolutnie)
RAD	Real	Promień okręgu (bez znaku)
STA1	Real	Kąt początkowy
INDA	Real	Przyrost kąta (ze znakiem)
FFD	Real	Posuw dla dosuwu na głębokość (ruch w osi dosuwowej)
FFP1	Real	Posuw dla obróbki powierzchni (ruch w płaszczyźnie)
MID	Real	Maksymalna głębokość dosuwu (bez znaku)

Przy pomocy tego cyklu obrabia się rowki podłużne (nazywane otworami podłużnymi, ang. *Long Hole*), które są usytuowane na okręgu. Dłuższa oś rowków jest usytuowana promieniowo. Szerokość rowka jest określana przez średnicę narzędzia (narzędzie w postaci freza walcowego wcina się tylko wzdłuż osi rowka bez wykonywania dodatkowych ruchów – Rys. 168). Wewnętrznie w cyklu jest obliczana optymalna droga ruchu narzędzia, co wyklucza niepotrzebne ruchy jałowe. Jeżeli dla obróbki rowka jest koniecznych wiele dosuwów na głębokość, wówczas dosuw

następuje na przemian w punktach końcowych. Tor będący do przebycia w płaszczyźnie wzdłuż osi podłużnej zmienia swój kierunek po każdym dosuwie. Cykl poszukuje samodzielnie najkrótszej drogi przy przejściu do następnego rowka.



Rys. 167. Cykl frezowania otworów podłużnych na okręgu LONGHOLE



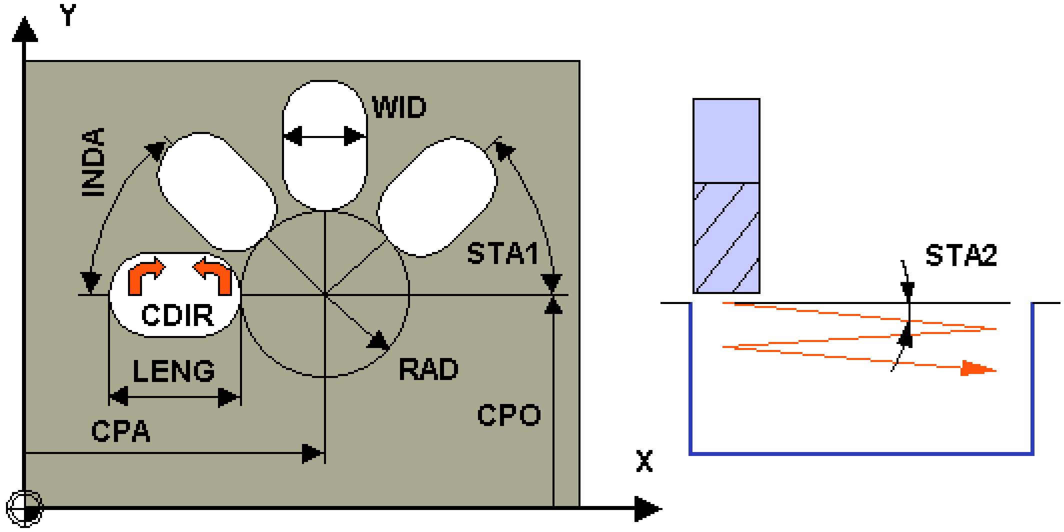
Rys. 168. Cykl LONGHOLE – schemat przebiegu obróbki

10.4.2. Rowki podłużne na okręgu – SLOT1

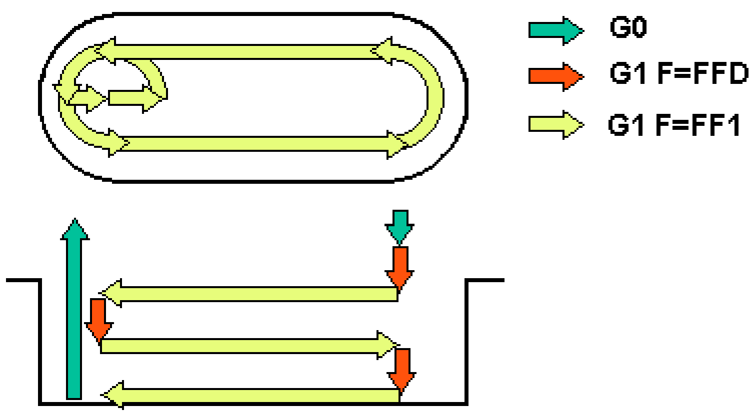
SLOT1(RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF, _FALD, _STA2) (Rys. 169)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość rowka (absolutnie)
DPR*	Real	Ostateczna głębokość rowka (przyrostowo, bez znaku)
NUM	Integer	Ilość rowków
LENG	Real	Długość rowków (bez znaku)
WID	Real	Szerokość rowków (bez znaku)
CPA	Real	Odcięta punktu środkowego (absolutnie)
CPO	Real	Odcięta punktu środkowego (absolutnie)
RAD	Real	Promień okręgu (bez znaku)
STA1	Real	Kąt początkowy
INDA	Real	Przyrost kąta (ze znakiem)
FFD	Real	Posuw dla dosuwu
FFP1	Real	Posuw dla obróbki powierzchni bocznej
MID	Real	Maksymalna głębokość jednego dosuwu (bez znaku)
CDIR	Integer	Kierunek obróbki: 2 (G2), 3 (G3)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej
VARI	Integer	Rodzaj obróbki: 0 – obróbka kompletna (zgrubna i wykańczająca) 1 – obróbka zgrubna z pozostawieniem naddatku 2 – tylko obróbka wykańczająca
MIDF	Real	Maksymalna głębokość dosuwu przy obróbce wykańczającej
FFP2	Real	Posuw przy obróbce wykańczającej
SSF	Real	Prędkość obrotowa przy obróbce wykańczającej
_FALD	Real	Naddatek na obróbkę wykańczającą na dnie rowka
_STA2	Real	Maksymalny kąt zagłębiania się narzędzia

SLOT1 jest cyklem do obróbki zgrubnej i/lub wykańczającej rowków podłużnych, rozmieszczonych na okręgu koła. Oś wzdłużna rowków jest ustawiona promieniowo. W przeciwieństwie do cyklu LONGHOLE jest podawana szerokość rowka (frez walcowy o mniejszej średnicy niż szerokość rowka wykonuje zarówno dosuwowy ruch wgłębny jak i ruchy w płaszczyźnie obróbki – Rys. 170).



Rys. 169. Cykl frezowania rowków podłużnych na okręgu SLOT1



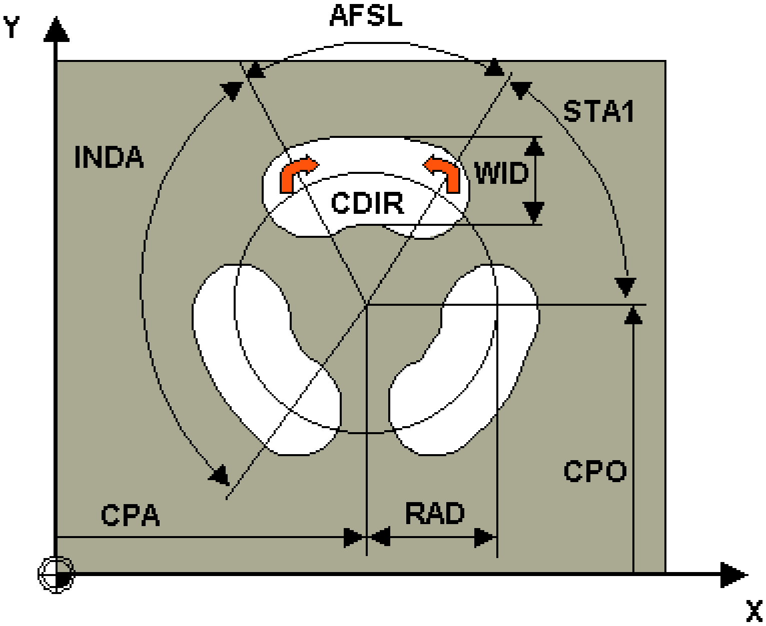
Rys. 170. Cykl SLOT1 – schemat przebiegu obróbki

10.4.3. Rowek kołowy na okręgu – SLOT2

SLOT2(RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF) (Rys. 171)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość rowka (absolutnie)
DPR*	Real	Ostateczna głębokość rowka (przyrostowo, bez znaku)
NUM	Integer	Ilość rowków
AFSL	Real	Kątowa długość rowków (bez znaku)
WID	Real	Szerokość rowka (bez znaku)
CPA	Real	Odcięta punktu środkowego (absolutnie)
CPO	Real	Odcięta punktu środkowego (absolutnie)
RAD	Real	Promień okręgu (bez znaku)
STA1	Real	Kąt początkowy
INDA	Real	Przyrost kąta (ze znakiem)
FFD	Real	Posuw dla dosuwu na głębokość
FFP1	Real	Posuw dla obróbki powierzchni bocznej
MID	Real	Maksymalna głębokość jednego dosuwu (bez znaku)
CDIR	Integer	Kierunek obróbki: 2 (G2), 3 (G3)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej
VARI	Integer	Rodzaj obróbki: 0 – obróbka kompletna (zgrubna i wykańczająca) 1 – obróbka zgrubna z pozostawieniem naddatku 2 – tylko obróbka wykańczająca
MIDF	Real	Maksymalna głębokość dosuwu przy obróbce wykańczającej
FFP2	Real	Posuw dla obróbki wykańczającej
SSF	Real	Prędkość obrotowa przy obróbce wykańczającej

SLOT2 jest cyklem do obróbki zgrubnej i/lub wykańczającej rowków o zarysie łukowym, umieszczonych na okręgu koła. Obróbka odbywa się frezem walcowym o średnicy mniejszej niż szerokość rowka. Funkcjonowanie analogiczne do cyklu SLOT1.



Rys. 171. Cykl frezowania otworów kołowych na okręgu SLOT2

10.4.4. Frezowanie kieszeni prostokątnej – POCKET1

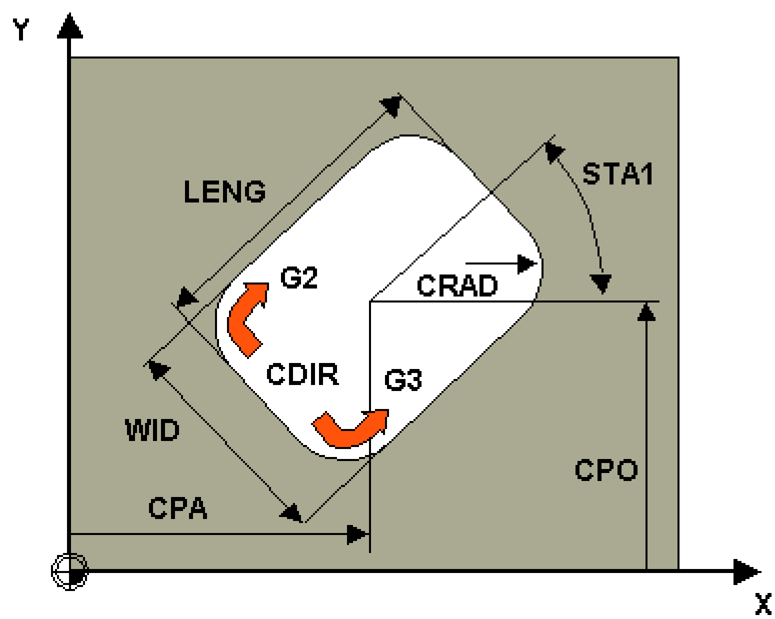
POCKET1(RTP, RFP, SDIS, DP, DPR, LENG, WID, CRAD, CPA, CPD, STA1, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF) (Rys. 172, Rys. 173)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość kieszeni (absolutnie)
DPR*	Real	Ostateczna głębokość kieszeni (przyrostowo, bez znaku)
LENG	Real	Długość kieszeni (bez znaku lub ze znakiem)
WID	Real	Szerokość kieszeni (bez znaku lub ze znakiem)
CRAD	Real	Promień narożnika (bez znaku)
CPA	Real	Odcięta punktu środkowego (absolutnie)
CPO	Real	Odcięta punktu środkowego (absolutnie)
STA1	Real	Kąt między osią wzdłużną i odciętą: 0° ≤ STA1 < 180°
FFD	Real	Posuw dla dosuwu
FFP1	Real	Posuw dla obróbki bocznej
MID	Real	Maksymalna głębokość dosuwu (bez znaku)
CDIR	Integer	Kierunek obróbki: 2 (G2), 3 (G3)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej
VARI	Integer	Rodzaj obróbki: 0 – obróbka kompletna (zgrubna i wykańczająca) 1 – obróbka zgrubna z pozostawieniem naddatku 2 – tylko obróbka wykańczająca
MIDF	Real	Maksymalna głębokość dosuwu dla obróbki wykańczającej
FFP2	Real	Posuw dla obróbki wykańczającej
SSF	Real	Prędkość obrotowa przy obróbce wykańczającej

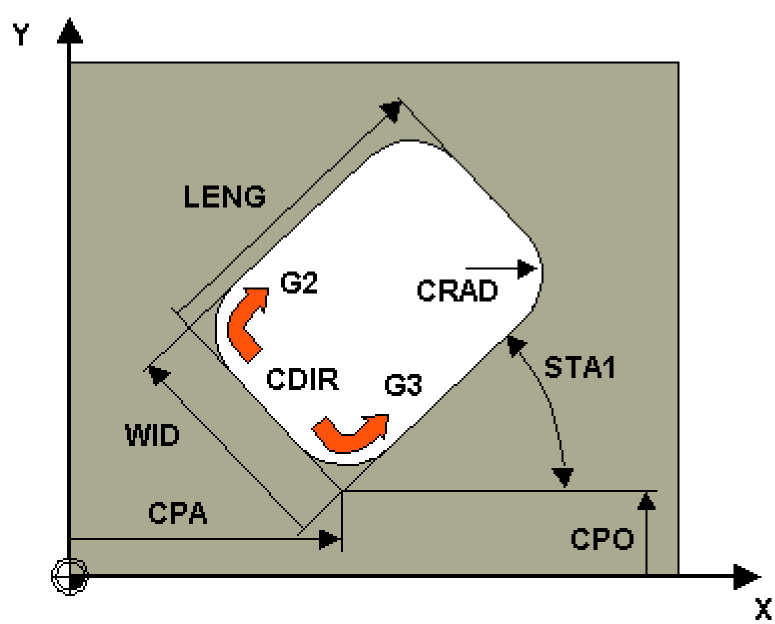
POCKET1 cyklem do obróbki zgrubnej i/lub wykańczającej kieszeni o zarysie prostokątnym przy pomocy freza walcowego. Przy pomocy tego cyklu można obrabiać kieszenie o dowolnym położeniu na płaszczyźnie obróbki. Sposób wymiarowania położenia kieszeni zależy od zmiennej nastawczej cykli _ZSD[2]. Dopuszczalne są dwie jej wartości:

_ZSD[2]=0 – położenie kieszeni względem jej środka, długość i szerokość podawane bez znaku (Rys. 172);

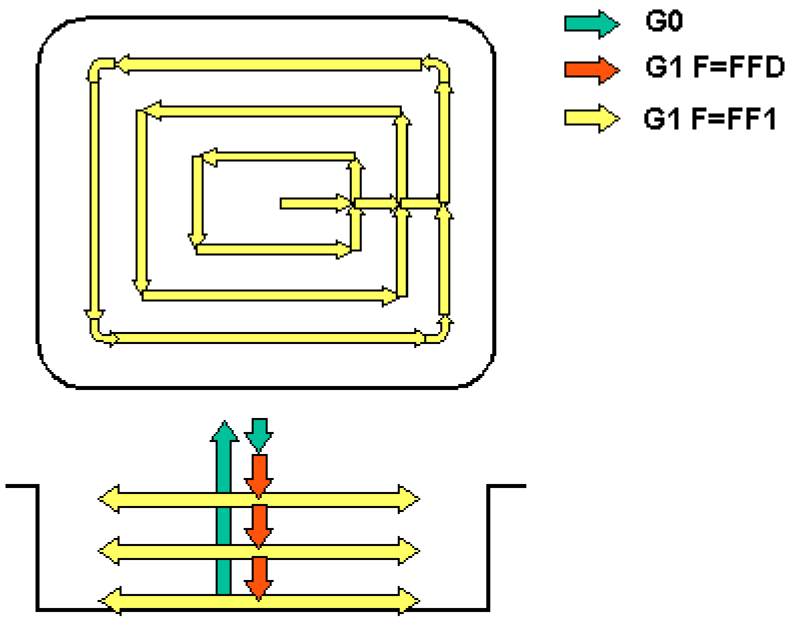
_ZSD[2]=1 – położenie kieszeni względem narożnika, długość i szerokość podawane ze znakiem (Rys. 173).



Rys. 172. Cykl frezowania kieszeni prostokątnej POCKET1 (_ZSD[2]=0)



Rys. 173. Cykl frezowania kieszeni prostokątnej POCKET1 (_ZSD[2]=1)

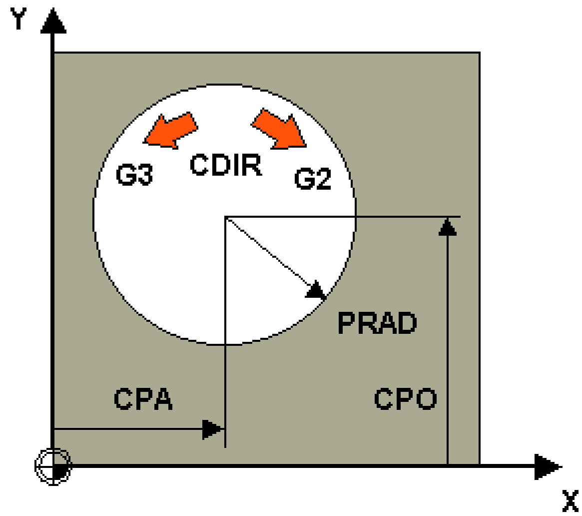


Rys. 174. Cykl POCKET1 – schemat przebiegu obróbki

10.4.5. Frezowanie kieszeni okrągłej – POCKET2

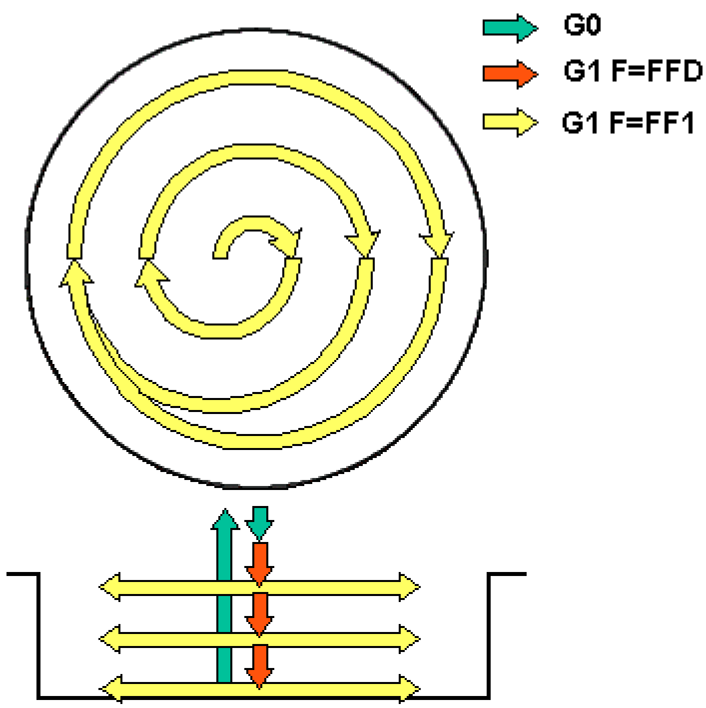
POCKET2(RTP, RFP, SDIS, DP, DPR,PRAD, CPA, CPO, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FF2, SSF) (Rys. 175)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Ostateczna głębokość kieszeni (absolutnie)
DPR*	Real	Ostateczna głębokość kieszeni (przyrostowo, bez znaku)
PRAD	Real	Promień kieszeni (bez znaku)
CPA	Real	Odcięta punktu środkowego (absolutnie)
CPO	Real	Odcięta punktu środkowego (absolutnie)
FFD	Real	Posuw dla dosuwu
FFP1	Real	Posuw dla obróbki bocznej
MID	Real	Maksymalna głębokość dosuwu (bez znaku)
CDIR	Integer	Kierunek obróbki: 2 (G2), 3 (G3)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej
VARI	Integer	Rodzaj obróbki: 0 => obróbka kompletna (zgrubna i wykańczająca) 1 => obróbka zgrubna z pozostawieniem naddatku 2 => tylko obróbka wykańczająca
MIDF	Real	Maksymalna głębokość dosuwu dla obróbki wykańczającej
FFP2	Real	Posuw dla obróbki wykańczającej
SSF	Real	Prędkość obrotowa przy obróbce wykańczającej



Rys. 175. Cykl frezowania kieszeni okrągłej POCKET2

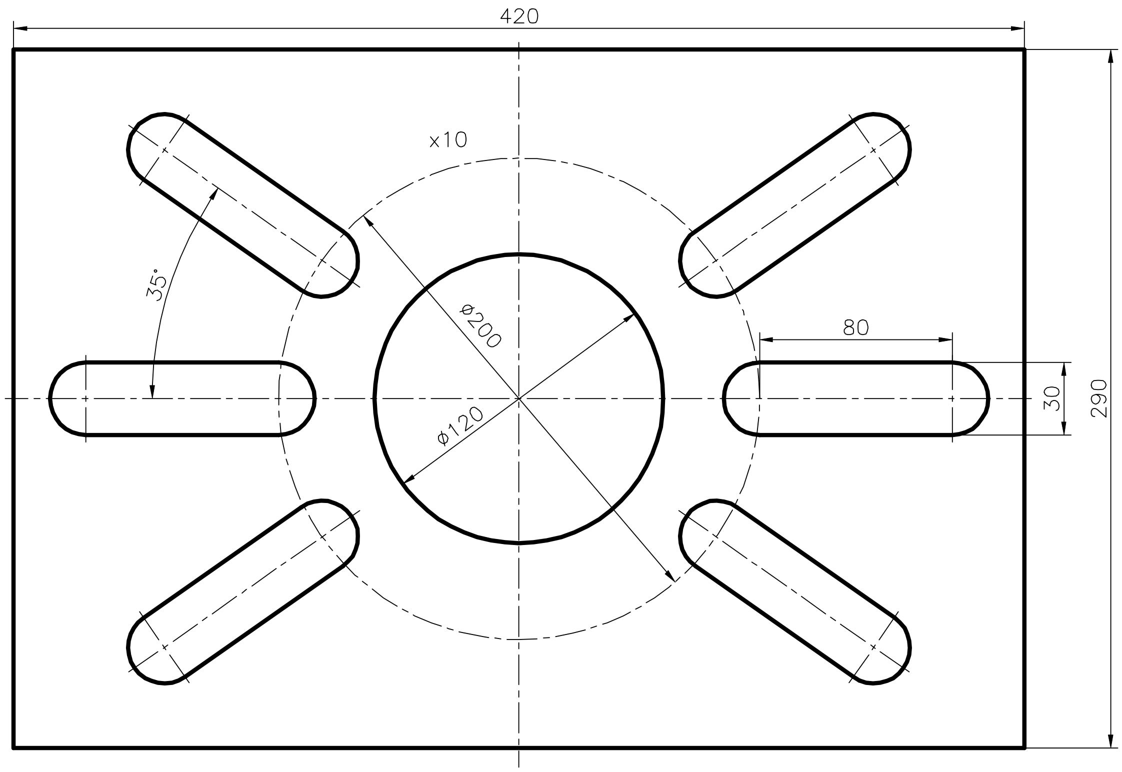
Działanie POCKET2 jest analogiczne jak cyklu POCKET1 z wyjątkiem innego kształtu obrabianej kieszeni i innej strategii obróbki (Rys. 176).



Rys. 176. Cykl POCKET2 – schemat przebiegu obróbki

10.4.6. Przykład

Zaprogramować obróbkę otworu i rowków w płycie pokazanej na Rys. 177 wykorzystując cykle frezowania kieszeni i rowków.



Rys. 177. Szkic przedmiotu do przykładu wykorzystania cykli obróbki rowków i kieszeni

Rozwiązanie:

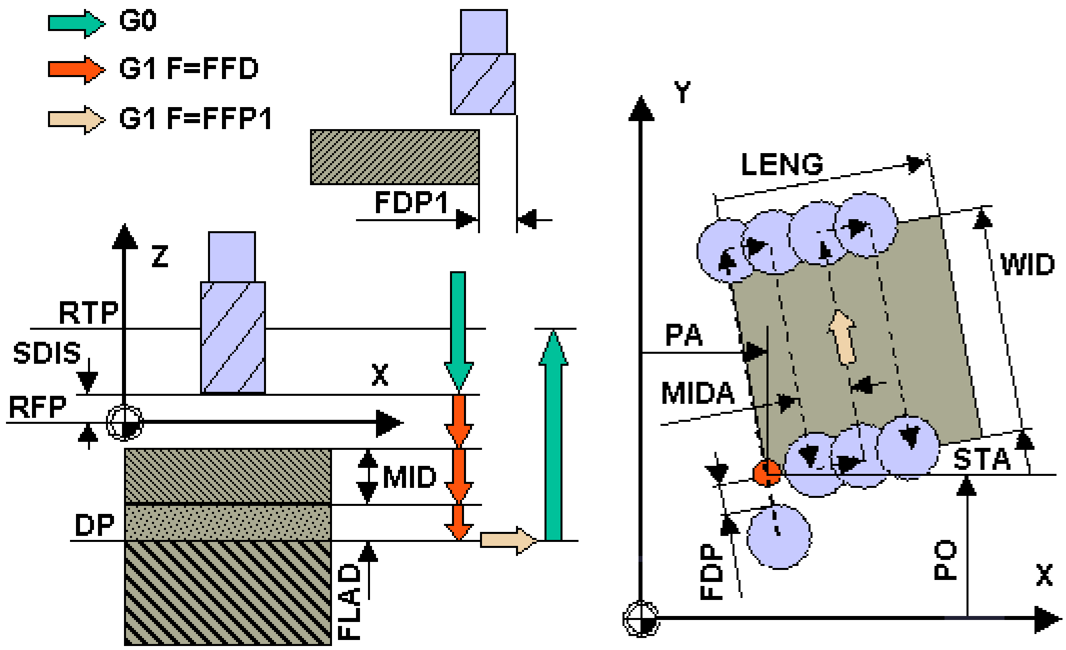
```
%_N_EX17_MPF
; 03-10-2003
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 MSG("FREZOWANIE KIESZENI")
N15 T1 D1 S650 F120 M3 M8 M6; FREZ D32
N20 POCKET2(100,0,3,-10,0,60,0,0,50,100,3,2,0.5,0,10,150,1000)
N25 MSG("FREZOWANIE ROWKOW")
N30 T2 D1 S900 F120 M3 M8 M6; FREZ D16
N35 SLOT1(100,0,2,-10,0,3,110,30,0,0,85,-35,35,50,100,3,2,0.5,0,10,150,1000,0,0)
N40 SLOT1(100,0,2,-10,0,3,110,30,0,0,85,145,35,50,100,3,2,0.5,0,10,150,1000,0,0)
N45 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N50 M30
```

10.4.7. Frezowanie płaszczyzny – CYCLE71

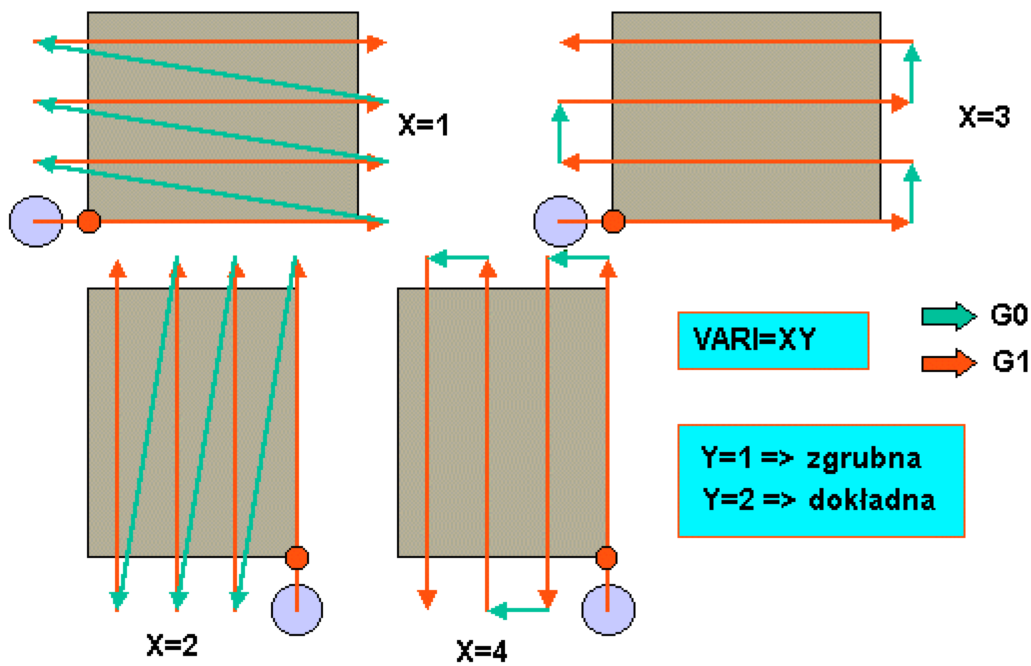
CYCLE71(RTP, RFP, SDIS, DP, PA, PO, LENG, WID, STA, MID, MIDA, FDP, FALD, FFP1, VARI, FDP1) (Rys. 178, Rys. 179)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP	Real	Ostateczna głębokość obróbki (absolutnie)
PA	Real	Odcięta punktu początkowego (absolutnie)
PO	Real	Rzędna punktu początkowego (absolutnie)
LENG	Real	Długość obszaru obróbki w osi odciętych (przyrostowo, ze znakiem)
WID	Real	Długość obszaru obróbki w osi rzędnych (przyrostowo, ze znakiem)
STA	Real	Kąt między osią wzdłużną obszaru obróbki i osią odciętych (bez znaku); zakres wartości: $0^\circ \leq STA < 180^\circ$
MID	Real	Maksymalna głębokość dosuwu (bez znaku)
MIDA	Real	Maksymalna szerokość dosuwu przy wybieraniu materiału w płaszczyźnie jako wartość (bez znaku)
FDP	Real	Droga odsunięcia w płaszczyźnie (przyrostowo, bez znaku)
FALD	Real	Naddatek na obróbkę wykańczającą na głębokości (przyrostowo, bez znaku)
FFP1	Real	Posuw dla obróbki powierzchni
VARI	Integer	Rodzaj obróbki: (bez znaku) MIEJSCE JEDNOSTEK – Wartości: 1 – obróbka zgrubna z pozostawieniem naddatku na obróbkę wykańczającą 2 – obróbka wykańczająca MIEJSCE DZIESIĄTEK – Wartości: 1 – równolegle do odciętej, w jednym kierunku 2 – równolegle do rzędnej, w jednym kierunku 3 – równolegle do odciętej, z kierunkiem zmiennym 4 – równolegle do rzędnej, z kierunkiem zmiennym
FDP1	Real	Wyjście w kierunku ustawionej płaszczyzny (przyrostowo, bez znaku)

CYCLE71 pozwala na frezowanie powierzchnią czołową freza (głowicy frezowej) powierzchni prostokątnej, dowolnie zorientowanej. Cykl rozróżnia obróbkę zgrubną – zbieranie materiału na powierzchni wieloma krokami (wieloma dosuwami wgłębnymi) z pozostawieniem naddatku na obróbkę wykańczającą lub obróbkę wykańczającą – jednokrotne przejście po powierzchni. Cykl nie uwzględnia korekty promienia narzędzia. Dosuw na kolejną głębokość obróbki jest wykonywany poza definiowanym obszarem obróbki.



Rys. 178. Cykl frezowania płaszczyzny CYCLE71

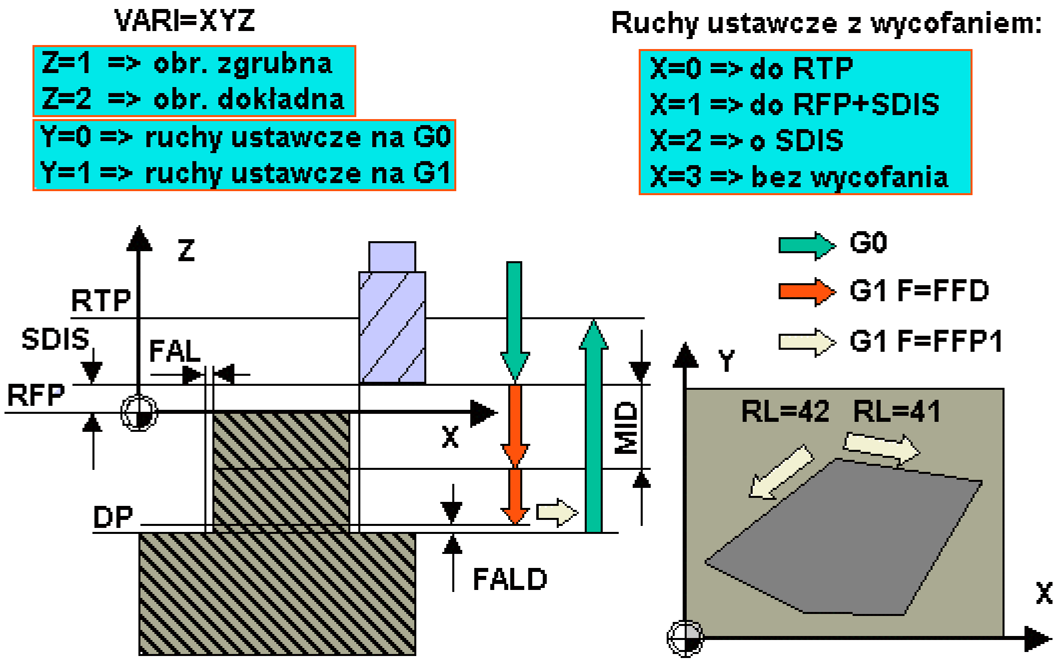


Rys. 179. Cykl frezowania płaszczyzny CYCLE71 (c.d.)

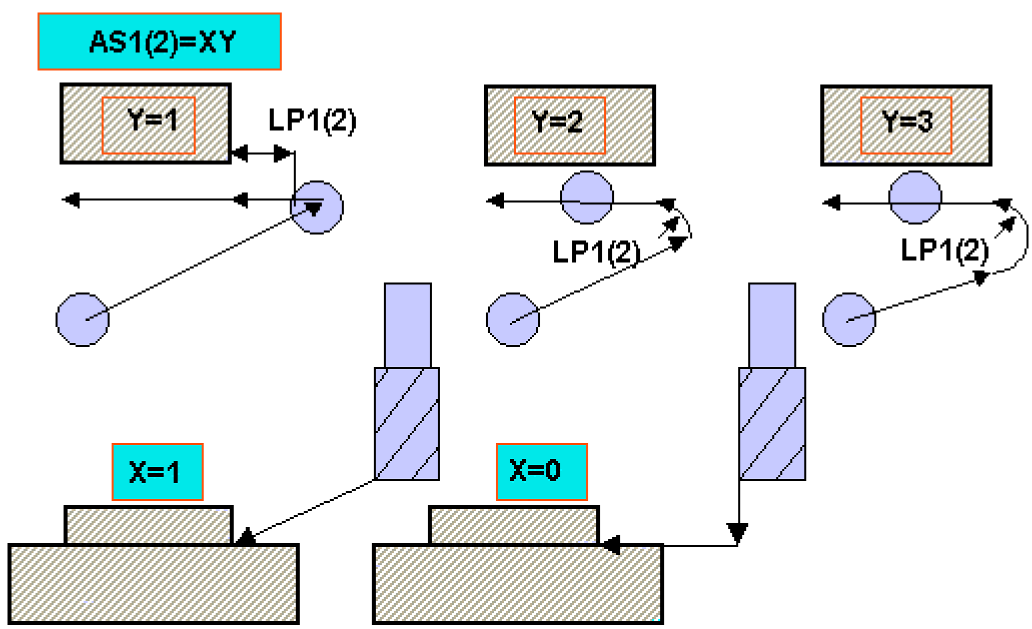
10.4.8. Frezowanie konturu – CYCLE72

CYCLE72(KNAME, RTP, RFP, SDIS, DP, MID, FAL, FALD, FFP1, FFD, VARI, RL, AS1, LP1, FF3, AS2, LP2) (Rys. 180, Rys. 181)

KNAME	String	Nazwa podprogramu obróbki konturu
RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP	Real	Głębokość obróbki (absolutnie)
MID	Real	Maksymalna głębokość dosuwu (przyrostowo, bez znaku)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej (bez znaku)
FALD	Real	Naddatek na obróbkę wykańczającą na dnie (bez znaku)
FFP1	Real	Posuw dla obróbki powierzchni (bez znaku)
FFD	Real	Posuw dla dosuwu na głębokość (bez znaku)
VARI	Integer	Rodzaj obróbki: MIEJSCE JEDNOSTEK – Wartości: 1 – obróbka zgrubna 2 – obróbka wykańczająca MIEJSCE DZIESIĄTEK – Wartości: 0 – drogi pośrednie przy pomocy G0 1 – drogi pośrednie przy pomocy G1 MIEJSCE SETEK – Wartości: 0 – wycofanie dla dróg pośrednich do RTP 1 – wycofanie dla dróg pośrednich do RFP+SDIS 2 – wycofanie w przypadku dla dróg pośrednich o SDIS 3 – nie ma wycofania w przypadku dróg pośrednich
RL	Integer	Kierunek kompensacji promienia narzędzia: 41 (G41), 42 (G42)
AS1	Integer	Specyfikacja drogi dosuwu: MIEJSCE JEDNOSTEK – Wartości: 1 – prosta styczna 2 – półkole 3 – ćwierć okręgu MIEJSCE DZIESIĄTEK – Wartości: 0 – dosunięcie do konturu na płaszczyźnie 1 – dosunięcie do konturu po torze przestrzennym
LP1	Real	Długość drogi dosuwu (w przypadku prostej) wzgl. promień toru punktu środkowego frezu (w przypadku okręgu, wprowadzić bez znaku)
FF3	Real	Posuw wycofania dla pozycjonowania pośredniego na płaszczyźnie (bez kontaktu z materiałem)
AS2	Integer	Specyfikacja drogi odsunięcia: MIEJSCE JEDNOSTEK – Wartości: 1 – prosta styczna 2 – półkole 3 – ćwierć okręgu MIEJSCE DZIESIĄTEK – Wartości: 0 – odsunięcie od konturu na płaszczyźnie 1 – odsunięcie od konturu po torze przestrzennym
LP2	Real	Długość drogi odsunięcia lub promień toru punktu środkowego freza



Rys. 180. Cykl frezowania konturu CYCLE72



Rys. 181. Cykl frezowania konturu CYCLE72 (c.d.)

CYCLE72 pozwala na programowanie obróbki wzdłuż dowolnego konturu, zdefiniowanego w podprogramie, z możliwością wielu przejść. Cykl uwzględnia kompensację promienia narzędzia. Kontur niekoniecznie musi być zamknięty, obróbka wewnętrzna lub zewnętrzna jest definiowana przez położenie korekty promienia narzędzia (po lewej albo po prawej od konturu). **Kontur musi być programowany w tym kierunku, w którym ma być obrabiany**, ponieważ podprogram konturu jest

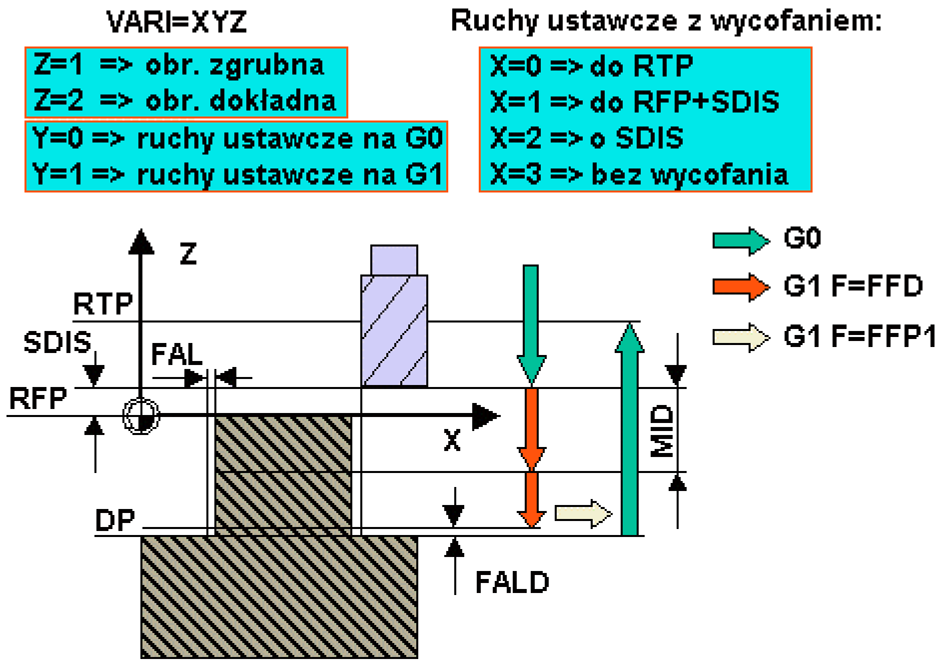
wywoływany bezpośrednio w cyklu. Przy programowaniu konturu należy przestrzegać następujących zasad:

- ➔ W podprogramie nie wolno przed pierwszą programowaną pozycją wybrać żadnego programowanego FRAME (TRANS, ROT, SCALE, MIRROR).
- ➔ Pierwszy blok podprogramu konturu jest blokiem prostej zawierającym G90, G0 i definiuje punkt startu konturu.
- ➔ Kompensacja promienia narzędzia jest włączana i wyłączana z cyklu nadrzędnego, dlatego w podprogramie konturu nie mogą być programowane adresy G40, G41, G42 .

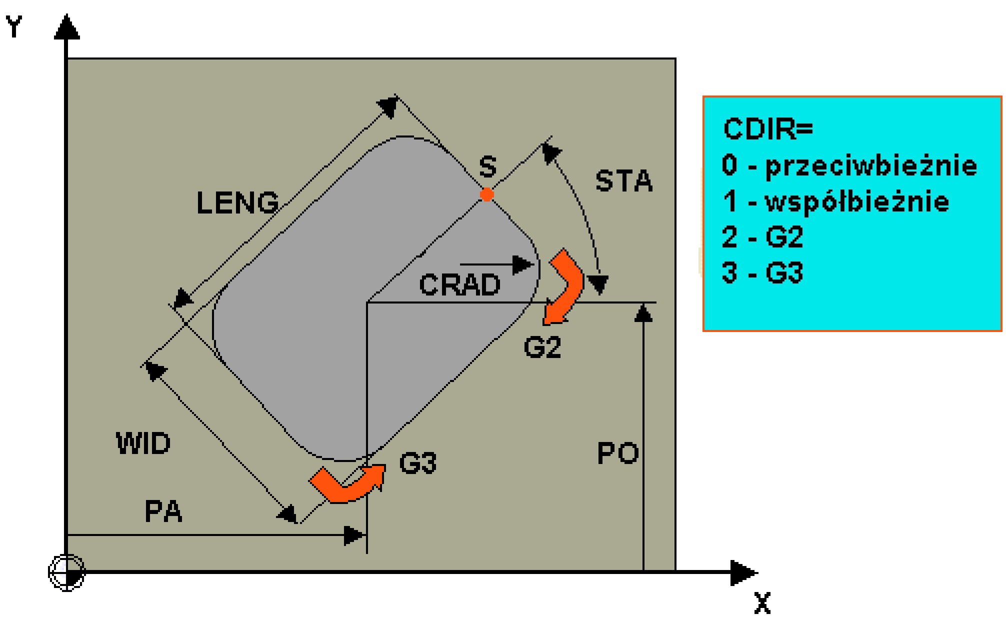
10.4.9. Frezowanie występu prostokątnego – CYCLE76

CYCLE76(RTP, RFP, SDIS, DP, DPR, LENG, WID, CRAD, PA, PO, STA,MID,FAL, FALD, FFP1, FFD, CDIR, VARI, AP1, AP2)
(Rys. 183, Rys. 184, Rys. 182)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Głębokość (absolutnie)
DPR*	Real	Głębokość (przyrostowo, bez znaku)
LENG	Real	Długość występu od narożnika (ze znakiem lub bez znaku)
WID	Real	Szerokość występu od narożnika (ze znakiem lub bez znaku)
CRAD	Real	Promień naroża (bez znaku)
PA	Real	Odcięta punktu bazowego (absolutnie)
PO	Real	Rzędna punktu bazowego (absolutnie)
STA	Real	Kąt między osią podłużną a odciętą
MID	Real	Głębokość dosuwu (przyrostowo, bez znaku)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej (bez znaku)
FALD	Real	Naddatek na obróbkę wykańczającą na dnie (bez znaku)
FFP1	Real	Posuw dla obróbki powierzchni bocznej
FFD	Real	Posuw dla dosuwu na głębokość (bez znaku)
CDIR	Integer	Kierunek obróbki: 0 – przeciwbieżny 1 – współbieżny 2 – G2 3 – G3
VARI	Integer	Rodzaj obróbki: 1 – zgrubna z pozostawieniem naddatku 2 – wykańczająca
AP1	Real	Długość przygotówki (bez znaku)
AP2	Real	Szerokość przygotówki (bez znaku)



Rys. 182. Cykl frezowania występu prostokątnego CYCLE76 i okrągłego CYCLE77



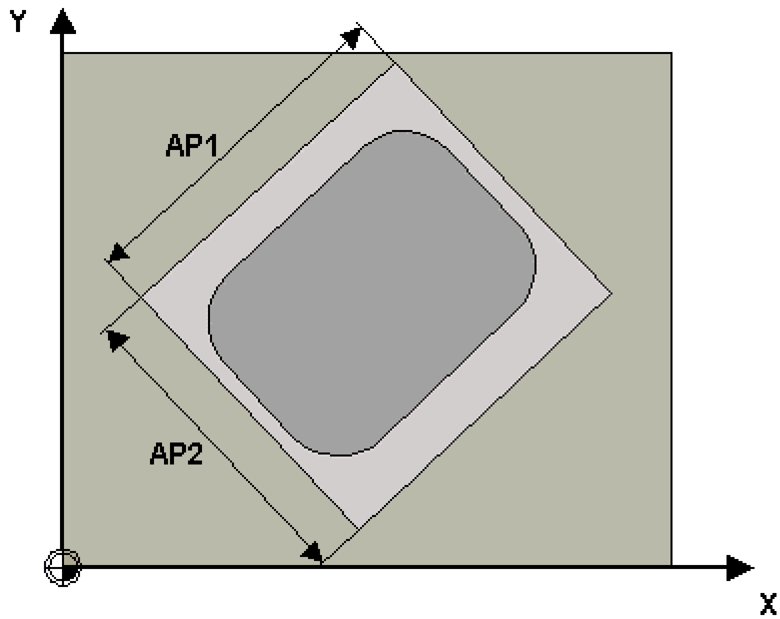
Rys. 183. Cykl frezowania występu prostokątnego CYCLE76 (c.d.)

CYCLE76 wykorzystuje opisany w poprzednim rozdziale cykl obróbki wzdłuż dowolnego konturu. Wewnętrznie definiuje taki kontur w postaci prostokątnego występu z zaokrąglonymi narożami i wywołuje wspomniany cykl CYCLE72. Podobnie jak to miało miejsce dl cyklu obróbki kieszeni prostokątnej POCKET1 (rozdz. 7.4.4) możliwe są dwa sposoby wymiarowania położenia występu:

_ZSD[2] = 0 – położenie występu względem jego środka, długość i szerokość podawane bez znaku (Rys. 183);

_ZSD[2] = 1 – położenie występu względem narożnika, długość i szerokość podawane ze znakiem (por. Rys. 173).

Punktem rozpoczęcia obróbki jest punkt na przecięciu dodatniej półosi odciętych (punkt **S** na Rys. 183) z uwzględnieniem dojścia po półokręgu (obróbka z kompensacją promienia narzędzia) oraz długości przygotówki.

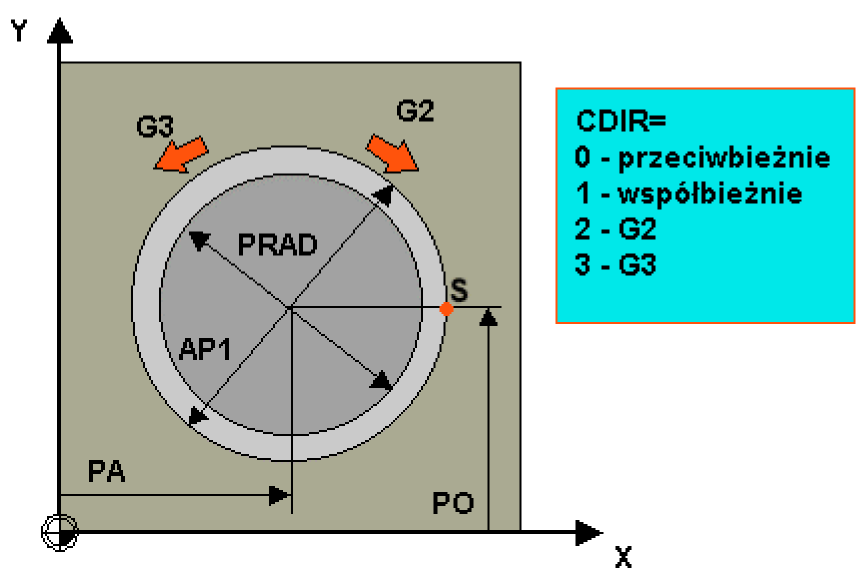


Rys. 184. Cykl frezowania występu prostokątnego CYCLE76 (c.d.)

10.4.10. Frezowanie występu okrągłego – CYCLE77

CYCLE77(RTP, RFP, SDIS, DP, DPR, PRAD, PA, PO, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, AP1) (Rys. 182, Rys. 185)

RTP	Real	Płaszczyzna wycofania (absolutnie)
RFP	Real	Płaszczyzna odniesienia (absolutnie)
SDIS	Real	Odstęp bezpieczeństwa (przyrostowo, bez znaku)
DP*	Real	Głębokość (absolutnie)
DPR*	Real	Głębokość (przyrostowo, bez znaku)
PRAD	Real	Średnica występu (bez znaku)
PA	Real	Odcięta punktu bazowego (absolutnie)
PO	Real	Rzędna punktu bazowego (absolutnie)
MID	Real	Głębokość dosuwu (bez znaku)
FAL	Real	Naddatek na obróbkę wykańczającą na powierzchni bocznej (bez znaku)
FALD	Real	Naddatek na obróbkę wykańczającą na dnie (bez znaku)
FFP1	Real	Posuw dla obróbki powierzchni bocznej
FFD	Real	Posuw dla dosuwu na głębokość
CDIR	Integer	Kierunek obróbki: 0 – przeciwbieżny 1 – współbieżny 2 – G2 3 – G3
VARI	Integer	Rodzaj obróbki: 1 – zgrubna z pozostawieniem naddatku 2 – wykańczająca
AP1	Real	Średnica przygotówki

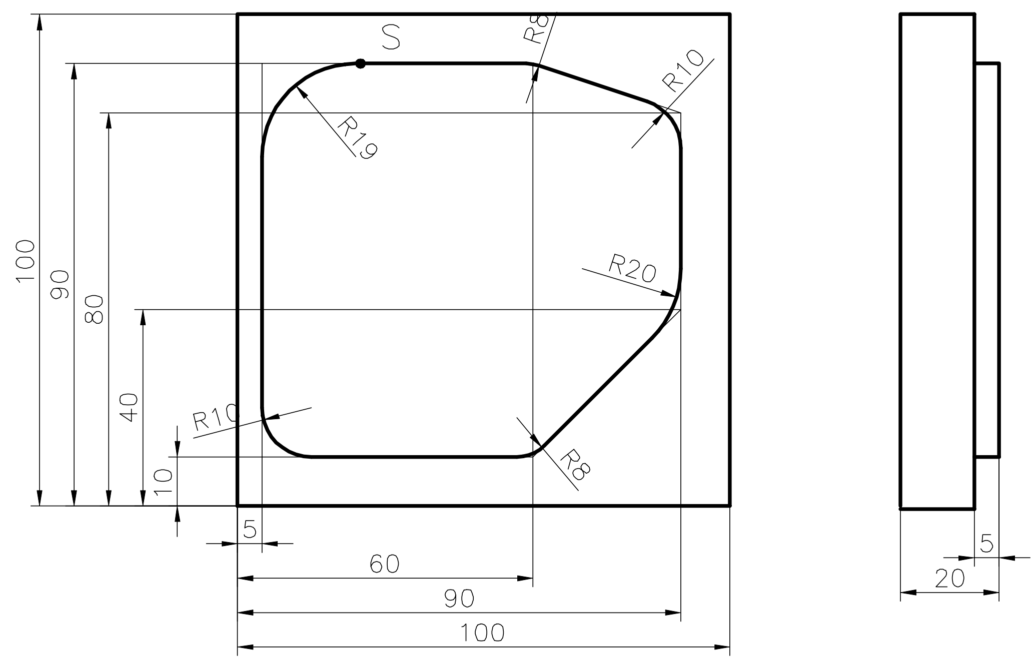


Rys. 185. Cykl frezowania występu okrągłego CYCLE77

Działanie i funkcjonowanie identyczne jak dla cyklu CYCLE76.

10.4.11. Przykład

Zaprogramować obróbkę konturu o kształcie podnym na Rys. 186 korzystając z cyklu obróbki konturu.



Rys. 186. Szkic przedmiotu do przykładu obróbki z wykorzystaniem cykli frezarskich (S – punkt rozpoczęcia obróbki w kierunku zgodnym z ruchem wskazówek zegara)

Rozwiązanie:

Podprogram definiujący kontur:

```
%_N_EX18_SPF
; KONTUR DO CYCLE72
N5 G1 X25 Y90
N10 G1 X60 Y90 RND=8
N15 X90 Y80 RND=10
N20 Y40 RND=20
N25 X60 Y10 RND=8
N30 X5 RND=10
N35 Y90 RND=19
N40 X25
N45 M17
```

Program sterujący:

```
%N_EX18_MPF
; 28-09-2004
N5 G40 G54 G71 G90 G94 DIAMOF KONT G450
N10 T1 D1 S1500 F250 M3 M8 M6
N15 G0 Z20
N20 X25 Y135
N25 CYCLE72("EX18",10,0,2,-5,3,0,0,200,50,201,41,3,5,250,3,5)
N30 G53 T0 D0 G0 X300 Y300 Z200 M9 M5
N35 M30
```

10.5. Cykle toczenia

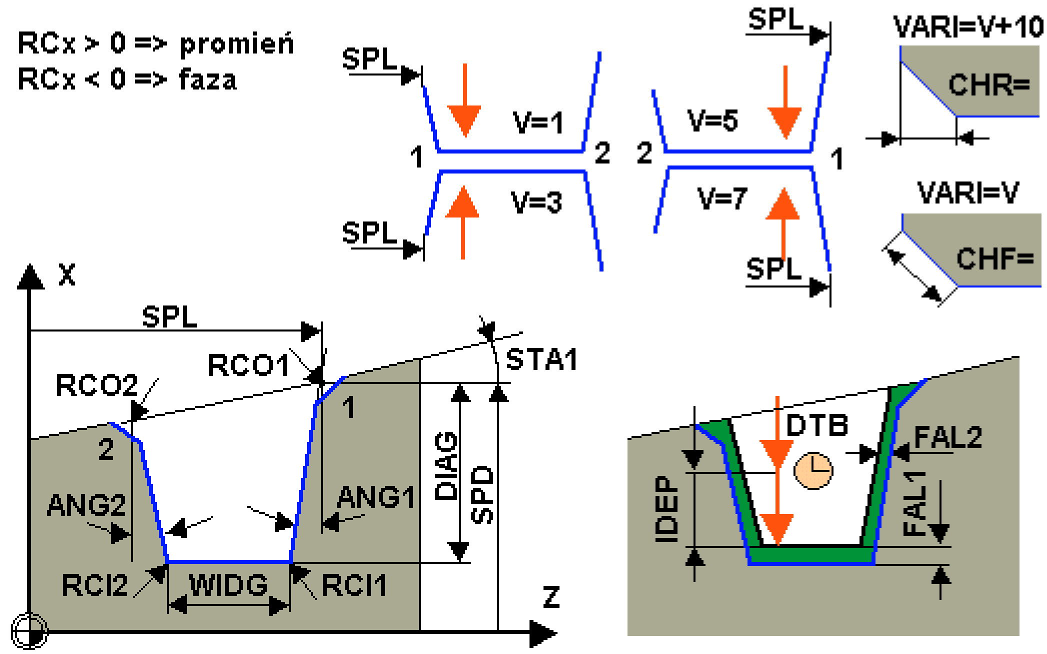
Cykle toczenia w większości układów sterowania obejmują najczęściej stosowane zabiegi: toczenia i wytaczania, toczenia rowków (kieszeni), toczenia gwintów czy toczenia podcięć obróbkowych. Dodatkowo występują tu cykle obróbki otworów, które jednak w układzie sterowania Sinumerik 810D/840D pochodzą z opisywanych w rozdz. 7.3 cykli wiercenia (za wyjątkiem specyficznych dla obróbki frezarskiej, np. macierzy otworów).

10.5.1. Toczenie rowków – CYCLE93

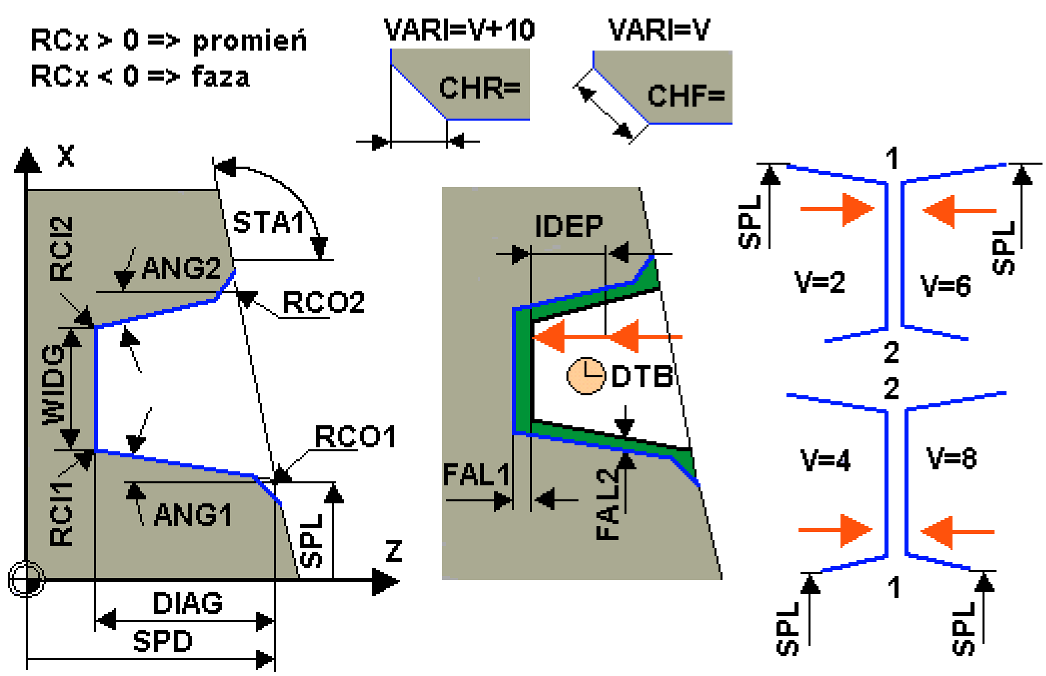
CYCLE93(SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI1, RCI2, FAL1, FAL2, IDEP, DTB, VARI) (Rys. 187**,** Rys. 188)

SPD	Real	Punkt początkowy w osi poprzecznej (bez znaku)
SPL	Real	Punkt początkowy w osi wzdłużnej
WIDG	Real	Szerokość rowka (bez znaku)
DIAG	Real	Głębokość rowka (bez znaku)
STA1	Real	Kąt między konturem i osią wzdłużną, 0° ≤ STA1 ≤ 180°
ANG1	Real	Kąt ściany po stronie punktu początkowego (bez znaku), 0° ≤ ANG1 < 89.999°
ANG2	Real	Kąt ściany po przeciwnej stronie punktu początkowego (bez znaku), 0° ≤ ANG2 < 89.999°
RCO1	Real	Promień/faza na zewnątrz po stronie punktu początkowego
RCO2	Real	Promień/faza na zewnątrz po przeciwnej stronie punktu początkowego
RCI1	Real	Promień/faza wewnątrz po stronie punktu początkowego
RCI2	Real	Promień/faza 2 wewnątrz po przeciwnej stronie punktu początkowego
FAL1	Real	Naddatek na obróbkę wykańczającą na dnie rowka
FAL2	Real	Naddatek na obróbkę wykańczającą na ścianach bocznych
IDEP	Real	Głębokość dosuwu (bez znaku)
DTB	Real	Postój czasowy na dnie rowka (wartość dodatnia w [s])
VARI	Integer	Kod rodzaju obróbki, dopuszczalne wartości 1÷8 i 11÷18

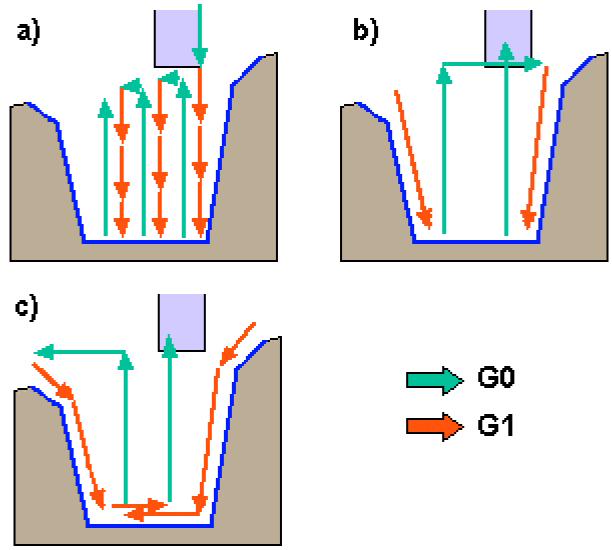
Toczenie rowków wykonuje się za pomocą wcinaków (poprzecznych lub wzdłużnych). Obróbka za ich pomocą odbywa się zasadniczo czołową krawędzią skrawającą, choć jeżeli mają być użyte również do obróbki wykańczającej to skrawać powinny także boczne krawędzie (na głębokość równą naddatkowi na dnie rowka – parametr FAL1). Strategię obróbki podczas toczenia rowków przedstawiono na Rys. 189. W pierwszej kolejności kolejnymi wcięciami na głębokość zadaną parametrem IDEP narzędzie wybiera zgrubnie naddatek (Rys. 189a). Następnie wyrównuje ściany boczne rowka (Rys. 189b) z pozostawieniem naddatku na obróbkę wykańczającą (parametr FAL2). W trzecim kroku obrabia kontur rowka na gotowo dwoma wejściami wzdłuż ścianek i dna rowka z uwzględnieniem faz i promieni zaokrągleń (Rys. 189c).



Rys. 187. Cykl toczenia rowków CYCLE93 (parametry dla rowków promieniowych)

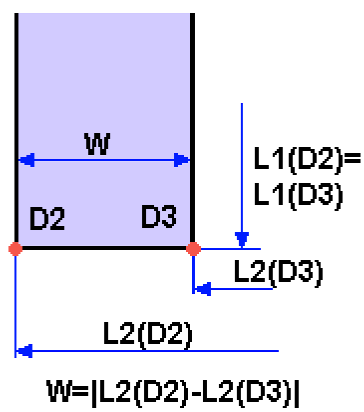


Rys. 188. Cykl toczenia rowków CYCLE93 (parametry dla rowków osiowych)



Rys. 189. Cykl toczenia rowków CYCLE93 – strategia obróbki: a) wybieranie zgrubne, b) wyrównanie zgrubne, c) obróbka wykańczająca

Jednym z warunków poprawnej obróbki z użyciem cyklu CYCLE93 jest odpowiednia deklaracja rejestrów narzędziowych. Na ich podstawie jest obliczana szerokość narzędzia przy wybieraniu zgrubnym, jak również są wymagane do automatycznej kompensacji promienia przy obróbce wyrównującej i wykańczającej. Do tego celu niezbędne jest zdefiniowanie **dwóch** rejestrów narzędziowych, przy czym wywołując cykl z rejestrem D*n* drugi z wymaganych rejestrów to D*n*+1. Jeżeli nie ma takiego rejestru bądź jest on zdefiniowany niepoprawnie to cykl zakończy się odpowiednim komunikatem błędu. Przykład poprawnie zdefiniowanych rejestrów narzędziowych pokazano na Rys. 190. Wynikowy wymiar **W** szerokości wcinaka jest obliczany jako wartość dodatnia różnicy odpowiednich wartości rejestrów (w zależności od kierunku rowka – promieniowego lub osiowego), stąd kolejność deklarowania naroży wcinaka nie ma istotnego znaczenia. Przy obróbce wykańczającej cykl automatycznie wykryje który rejestr opisuje które naroże (prawe-lewe; zewnętrzne-wewnętrzne) i dobierze je odpowiednio przy przejściu wzdłuż ścian rowka.

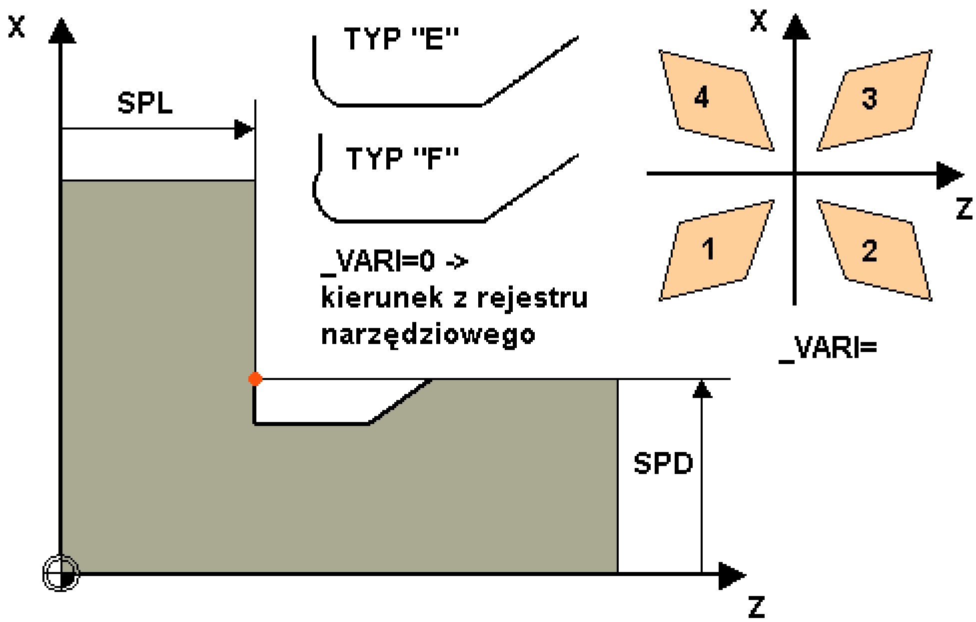


Rys. 190. Przykład deklaracji rejestrów narzędziowych dla potrzeb cyklu CYCLE93

10.5.2. Toczenie podcięć obróbkowych – CYCLE94

CYCLE94(SPD, SPL, FORM,_VARI) (Rys. 191)

SPD	Real	Położenie punktu bazowego w osi poprzecznej (bez znaku)
SPL	Real	Położenie punktu bazowego w osi wzdłużnej
FORM	Char	Definicja kształtu: E lub F
_VARI*	Integer	Specyfikacja kierunku narzędzia i położenia podcięcia: 0 – kierunek ostrza z rejestrów narzędziowych 1,2,3,4 – określony kierunek ostrza



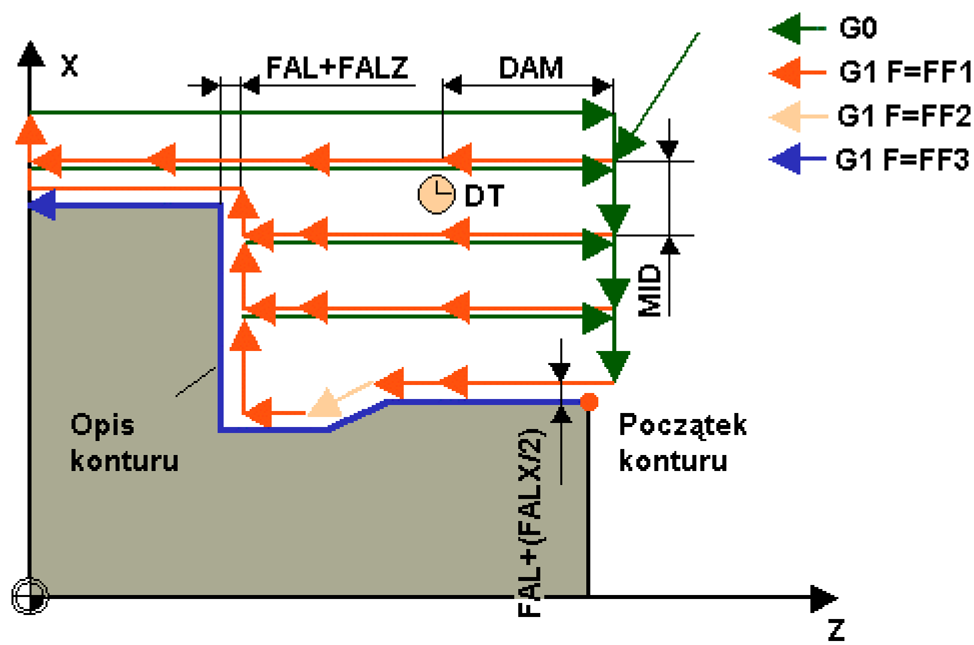
Rys. 191. Cykl toczenia podcięć obróbkowych CYCLE94

Podcięcia obróbkowe, wykonywane w cyklu CYCLE94, są oznaczane i wymiarowane wg normy DIN. Odpowiadają one polskiej normie M-02043, gdzie podcięciu typu E (wcięcie w powierzchnię walcową) odpowiada typ A, a typu F (wcięcie w powierzchnię walcową i czołową) – typ B. Rodzaj wykonywanego podcięcia (zewnętrzne-wewnętrzne; prawe-lewe) zależy również od kierunku ostrza stąd należy pamiętać o starannym zdefiniowaniu rejestrów narzędziowych.

10.5.3. Cykl toczenia i wytaczania – CYCLE95

CYCLE95(NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM)
(Rys. 192, Rys. 193)

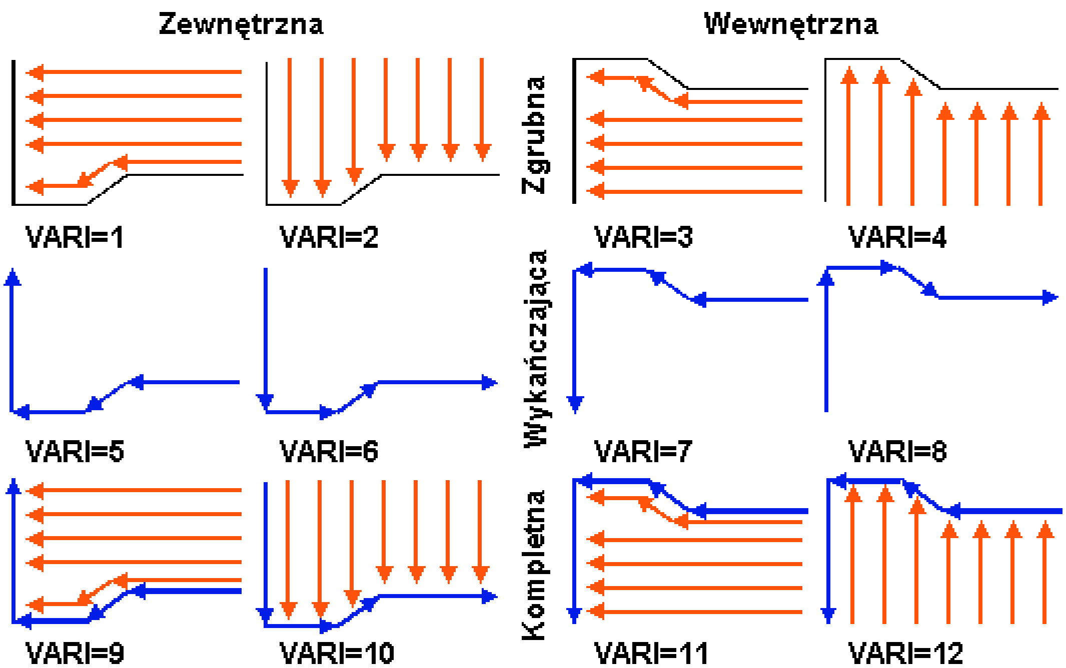
NPP	String	Nazwa podprogramu definicji konturu
MID	Real	Głębokość dosuwu (bez znaku)
FALZ*	Real	Naddatek na obróbkę wykańczającą w osi wzdłużnej
FALX*	Real	Naddatek na obróbkę wykańczającą w osi poprzecznej
FAL*	Real	Naddatek na obróbkę wykańczającą
FF1	Real	Posuw dla obróbki zgrubnej bez podcięcia
FF2	Real	Posuw dla zagłębiania się w elementy podcięcia
FF3	Real	Posuw dla obróbki wykańczającej
VARI	Integer	Rodzaj obróbki, zakres wartości 1÷12
DT	Real	Czas postoju przy łamaniu wióra
DAM	Real	Długość drogi skrawania do łamania wióra



Rys. 192. Cykl toczenia i wytaczania CYCLE95

CYCLE95 jest jednym z najbardziej złożonych cykli obróbkowych. Jego zadaniem jest toczenie zgrubne i/lub wykańczające powierzchni zewnętrznych lub wewnętrznych o złożonym kształcie, ograniczonych konturem zdefiniowanym w postaci podprogramu. Obróbka zgrubna jest realizowana za pomocą szeregu równoległych przejść (promieniowych lub osiowych) z posuwem zadanym parametrem FF1, których punkt końcowy jest wyznaczany na podstawie przebiegu konturu. Jeżeli kontur definiuje występowanie kieszeni to wcinanie się jest realizowane z posuwem zadanym parametrem FF2. Po obróbce zgrubnej jest pozostawiany naddatek na obróbkę wykańczającą. Może on być definiowany osobno w kierunku osiowym i promieniowym (parametry FALX i FALZ), globalnie dla wszystkich powierzchni (parametr FAL) lub jednocześnie obiema metodami. Obróbka

wykańczająca polega z kolei na wywołaniu podprogramu konturu z ustawieniem posuwu zadanego parametrem FF3 oraz automatyczną kompensacją promienia narzędzia.



Rys. 193. Cykl toczenia i wytaczania CYCLE95 – definicja rodzaju obróbki

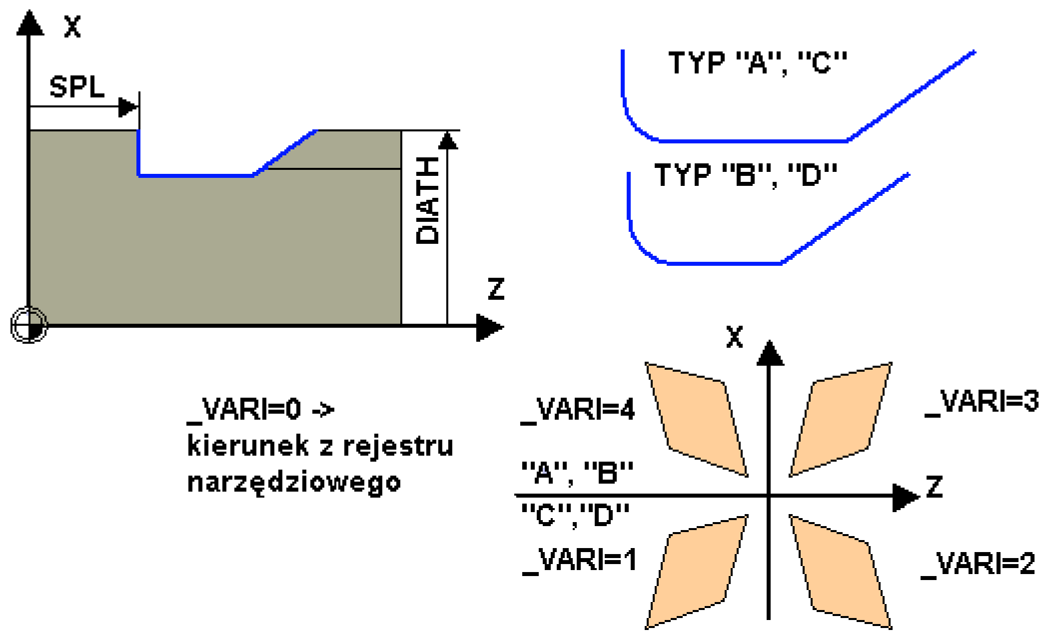
Istotne znaczenie ma dobór wartości parametru VARI, definiującego rodzaj obróbki (Rys. 193). Wyróżnić tu można obróbkę zewnętrzną lub wewnętrzną, zgrubną, wykańczającą lub kompletną (zgrubna i wykańczającą) oraz osiową lub promieniową. Łącznie zestawienie wszystkich rodzajów obróbki daje 12 jej odmian, kodowanych za pomocą parametru VARI.

Szczególną uwagę należy zwrócić na przygotowanie podprogramu definiującego kontur obrabiany. Musi on zawierać ciąg co najmniej trzech elementów typu linia lub łuk, definiowanych za pomocą standardowych funkcji interpolacji (G1, G2, G3...) z uwzględnieniem faz i zaokrągleń (adresy RND, RNDM, CHF, CHR). Pierwszy z bloków geometrycznych definiuje punkt początkowy konturu, stąd można w nim użyć interpolacji punktowej G0. Użycie innych rodzajów interpolacji jest niedopuszczalne. W podprogramie nie można wykorzystywać automatycznej kompensacji promienia narzędzia (G41, G42) – odpowiedni rodzaj jej kompensacji zostanie automatycznie włączony przez cykl. W podprogramie nie powinno się również stosować parametrów technologicznych (adresy S, F) ani narzędziowych (adresy T, D). Maksymalna liczba elementów konturu, jakie występują w podprogramie, jest ograniczona wewnętrznymi rozmiarami tablic je przechowującymi. Można przyjąć, że jest to ok. 50 elementów.

10.5.4. Toczenie podcięcia gwintu – CYCLE96

CYCLE96(DIATH, SPL, FORM, _VARI) (Rys. 194)

DIATH	Real	Nominalna średnica gwintu
SPL	Real	Punkt początkowy konturu w osi wzdłużnej
FORM	Char	Rodzaj podcięcia – A÷D
_VARI*	Integer	Specyfikacja kierunku narzędzia i położenia podcięcia: 0 – kierunek ostrza z rejestrów narzędziowych 1,2,3,4 – określony kierunek ostrza



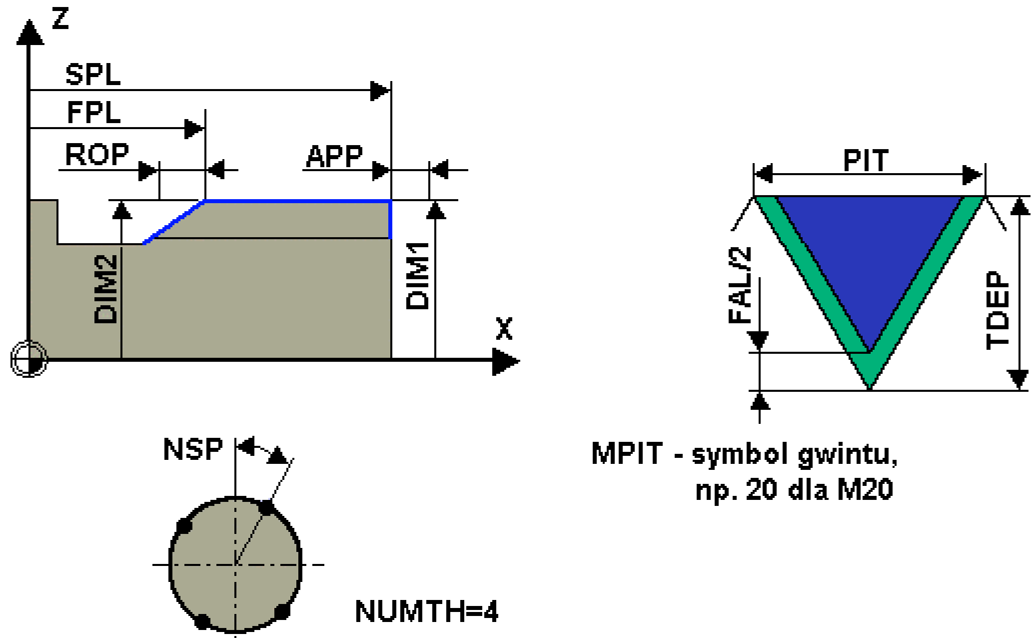
Rys. 194. Cykl toczenia podcięcia gwintu CYCLE96

CYCLE96 służy do wykonywania podcięć gwintu (przestrzeń dla wybiegu/dobiegu noża do nacinania gwintu) według normy niemieckiej DIN76 dla gwintów metrycznych (od M3 do M68). Jego funkcjonowanie jest zbliżone do opisanego wcześniej cyklu podcięć obróbkowych CYCLE94 (rozdz. 7.5.2).

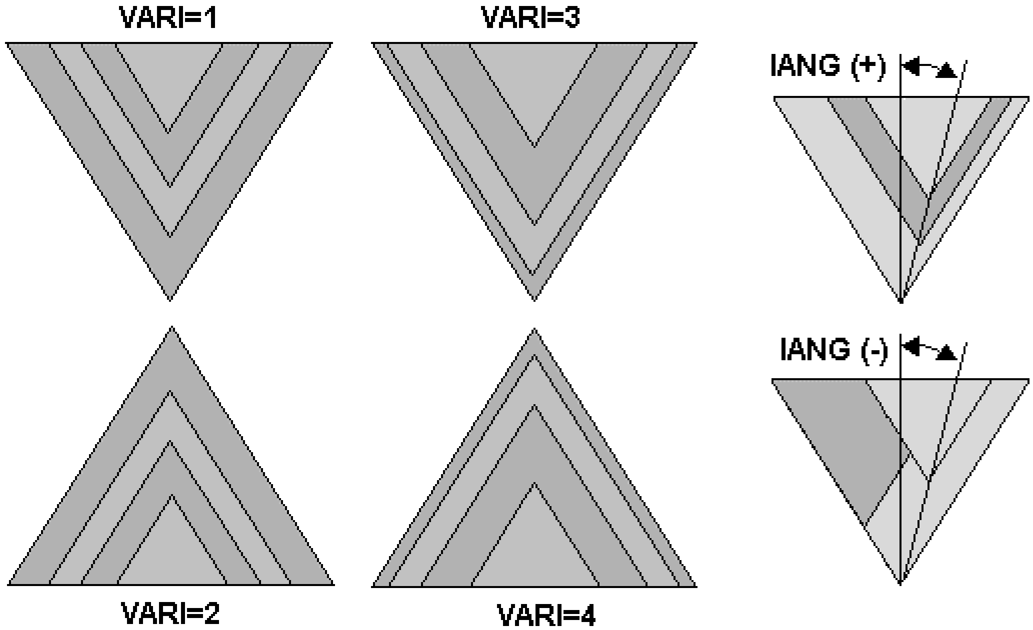
10.5.5. Toczenie gwintu prostego – CYCLE97

CYCLE97(PIT, MPIT, SPL, FPL, DM1, DM2, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, VARI, NUMTH) (Rys. 195, Rys. 196)

PIT	Real	Skok gwintu jako wartość (bez znaku)
MPIT	Real	Skok gwintu jako wielkość gwintu metrycznego: 3 (dla M3) ÷ 60 (dla M60)
SPL	Real	Punkt początkowy gwintu w osi wzdłużnej
FPL	Real	Punkt końcowy gwintu w osi wzdłużnej
DM1	Real	Średnica gwintu w punkcie początkowym
DM2	Real	Średnica gwintu w punkcie końcowym
APP	Real	Dobieg narzędzia (bez znaku)
ROP	Real	Wybieg narzędzia (bez znaku)
TDEP	Real	Głębokość zarysu gwintu (bez znaku)
FAL	Real	Naddatek na obróbkę wykańczającą (bez znaku)
IANG	Real	Kąt dosuwu przy zagłębianiu. Zakres wartości: Dodatni – dla dosuwu wzdłuż jednego kierunku Ujemny – dla dosuwu naprzemiennego
NSP	Real	Przesunięcie kątowe punktu startowego dla pierwszego zwoju gwintu (bez znaku)
NRC	Integer	Ilość przejść zgrubnych
NID	Integer	Ilość przejść wykańczających
VARI	Integer	Określenie rodzaju obróbki gwintu, wartości 1÷4: 1, 2 – stały dosuw (zewnętrzny, wewnętrzny) 3, 4 – stały przekrój warstwy skrawanej (zewnętrzny, wewnętrzny)
NUMTH	Integer	Ilość zwojów gwintu



Rys. 195. Cykl toczenia gwintu prostego CYCLE97



Rys. 196. Cykl toczenia gwintu – definicja rodzaju obróbki

Jak już wspomniano wcześniej (rozdz. 3.1) obróbka gwintów na tokarkach wymaga zaprogramowania wielu przejść narzędzia. Aby uniknąć żmudnego obliczania i programowania prostych ruchów najczęściej wykorzystuje się cykle. Omawiany CYCLE97 umożliwia obróbkę gwintów walcowych, stożkowych a także zarysów spiralnych na powierzchniach czołowych, gwintów wewnętrznych i zewnętrznych, jedno- i wielozwojnych, o dowolnych zarysach, wykonywanych przez płytki o pojedynczym występie lub wielowystępowe (tylko dla gwintów jednozwojnych). Dla gwintów metrycznych przewidziano wewnętrzną bazę danych o skokach gwintów, zadawanych w formie symbolicznej (parametr MPIT).

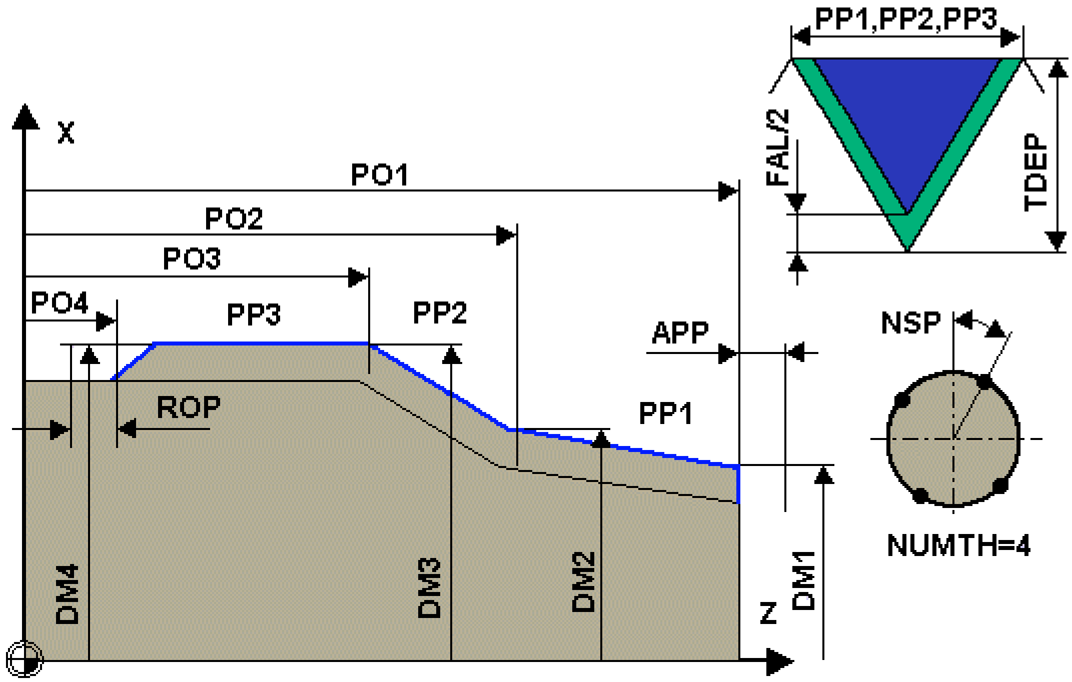
Obróbka gwintu prostego obejmuje NRC przejść zgrubnych (z metodą wcinania się narzędzia określoną parametrami VARI, IANG, TDEP, z pozostawieniem naddatku FAL; oraz NID przejść wykańczających, tj. po średnicy rdzenia gwintu. Mają one na celu wygładzenie powierzchni bruzd, powinno się zadać minimum 2÷3 przejścia wykańczające.

10.5.6. Toczenie gwintu złożonego – CYCLE98

CYCLE98(PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMTH)
(Rys. 197, Rys. 196)

PO1	Real	Punkt początkowy gwintu w osi wzdłużnej
DM1	Real	Średnica gwintu w punkcie początkowym
PO2	Real	Pierwszy punkt pośredni w osi wzdłużnej
DM2	Real	Średnica w pierwszym punkcie pośrednim
PO3	Real	Drugi punkt pośredni w osi wzdłużnej
DM3	Real	Średnica w drugim punkcie pośrednim
PO4	Real	Punkt końcowy gwintu w osi wzdłużnej
DM4	Real	Średnica w punkcie końcowym
APP	Real	Dobieg narzędzia (bez znaku)
ROP	Real	Wybieg narzędzia (bez znaku)
TDEP	Real	Głębokość zarysu gwintu (bez znaku)
FAL	Real	Naddatek na obróbkę wykańczającą
IANG	Real	Kąt dosuwu przy zagłębianiu. Zakres wartości: Dodatni – dla dosuwu wzdłuż jednego kierunku Ujemny – dla dosuwu naprzemiennego
NSP	Real	Przesunięcie kątowe punktu startowego dla pierwszego zwoju gwintu (bez znaku)
NRC	Integer	Ilość przejść zgrubnych
NID	Integer	Ilość przejść wykańczających
PP1	Real	1. skok gwintu jako wartość (bez znaku)
PP2	Real	2. skok gwintu jako wartość (bez znaku)
PP3	Real	3. skok gwintu jako wartość (bez znaku)
VARI	Integer	Określenie rodzaju obróbki gwintu, wartości 1÷4: 1, 2 – stały dosuw (zewnętrzny, wewnętrzny) 3, 4 – stały przekrój warstwy skrawanej (zewnętrzny, wewnętrzny)
NUMTH	Integer	Ilość zwojów gwintu

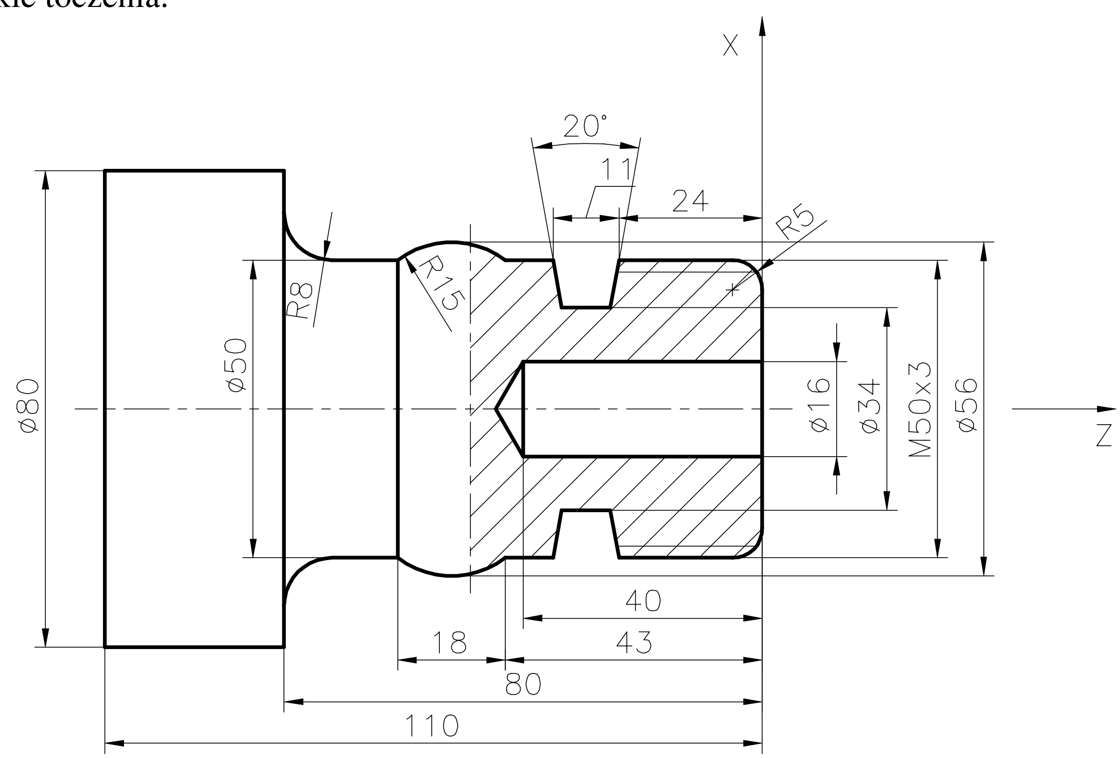
Gwint złożony to ciąg maksymalnie trzech odcinków gwintu prostego, tworzący jedną całość. Jako przykład wykorzystania cyklu CYCLE98 można podać toczenie tzw. gwintu gubionego, tj. zakończenie gwintu najczęściej walcowego odcinkiem gwintu stożkowego o tym samym skoku, przy czym wartość średnicy końcowej gwintu jest równa średnicy zewnętrznej przedmiotu przed gwintowaniem. W takim przypadku konieczne jest zdefiniowanie dwóch odcinków gwintu w cyklu CYCLE98. Sposób obróbki i znaczenie parametrów jest podobne jak w omawianym wcześniej cyklu gwintów prostych CYCLE97.



Rys. 197. Cykl toczenia gwintu złożonego CYCLE98

10.5.7. Przykład

Zaprojektować obróbkę przedmiotu przedstawionego na Rys. 198 wykorzystując cykle toczenia.



Rys. 198. Szkic przedmiotu do przykładu obróbki tokarskiej z wykorzystaniem cykli

Rozwiązanie:

```
%_N_EX27_MPF
; 26-10-2003
N5 G40 G54 G71 G90 G95 DIAMON KONT G450
N10 MSG("WIERCENIE OTWORU D16")
N15 T2 D1 S350 F0.25 M3 M8
N20 G0 X0 Z10
N25 CYCLE83(200,0,3,-44.48,,-25,,2,1,,1,0,,,,,)
N30 G0 X400 M5 M9
N35 MSG("TOCZENIE ZGRUBNE KONTURU")
N40 G96 T1 D1 S120 F0.2 M3 M8
N45 G0 X90 Z5
N50 CYCLE95("EX27",3,,,0.4,0.2,0.15,0.1,1,1,40,)
N55 G0 X400 Z200 M5 M9
N60 MSG("TOCZENIE DOKLADNE KONTURU")
N65 T3 D1 S100 F0.1 M3 M8
N70 G0 X60 Z-63
N75 G1 X50.8
N80 Z-79.5 RND=7.6
N85 X68
N90 G0 Z5
N95 X14
N100 G42 Z2
N105 EX27
N140 G40 G0 X90
N145 X400 Z200 M5 M9
N150 MSG("TOCZENIE ROWKA")
N155 G95 T5 D1 S900 F0.15 M3 M8
N160 G0 X60 Z-20
N165 CYCLE93(50,-24,8.178,8,0,10,10,0,0,0,0,0.1,0.1,4,1,5)
N170 G0 X400 Z200 M5 M9
N175 MSG("TOCZENIE GWINTU M50x3")
N180 T7 D1 S700 M3 M8
N185 G0 X60 Z-28
N190 CYCLE97(3,,-24,0,50,50,4,2,1.97,0.05,28,0,10,2,3,1,0)
N195 G53 T0 D0 G0 Z300 X300 M9 M5
N200 M30
```

Wykorzystany w programie podprogram definicji konturu zewnętrznego ma następującą postać:

```
%_N_EX27_SPF
; 26-10-2003
N5 G0 X14 Z0
N10 G1 X50 RND=5
N15 Z-43
N20 G3 G91 Z-18 CR=15
N25 G1 G90 Z-80 RND=8
N30 X80
N35 M17
```

11. Praktyczne aspekty przygotowania programów sterujących

11.1. Analiza przestrzeni roboczej

(W przygotowaniu)

11.2. Transmisja programów do układu sterowania

Ostatnim etapem projektowania programów sterujących (poza obrabiarką) jest ich wprowadzenie do układu sterowania. Współczesne układy sterowania oferują wiele różnych interfejsów, pozwalających na wymianę danych z otoczeniem, występujących również standardowo w komputerach PC – Ethernet, USB, PC-Card, FDD itp. Jednak jednym z podstawowych interfejsów, występującym również w układach nieco starszych, jest łącze szeregowe RS 232C. Łącze szeregowe powstało w 1962 r. na potrzeby obsługi modemów. W roku 1969 oficjalnie wprowadzono standard RS 232, zaś w 1986 – RS 232C.

Interfejs ten coraz rzadziej jest wykorzystywany w komputerach PC z uwagi na ograniczoną szybkość transmisji (najczęściej używane sterowniki pozwalają na szybkość transmisji ok. 115 kb/s), jednak w odniesieniu do sterowników obrabiarek CNC ta szybkość jest wystarczająca (w układach sterowania maksymalna szybkość transmisji na ogół jest dużo niższa, np. 9600 b/s). Często interfejs RS232C jest wykorzystywany również do komunikacji z urządzeniami kontrolno-pomiarowymi, miernikami cyfrowymi itp. Zaletą łącza szeregowego jest prostota realizacji transmisji (do łączenia dwukierunkowego w najprostszej postaci wystarczą trzy przewody), łatwość sterowania jej parametrami (muszą one być identyczne po stronie nadajnika i odbiornika) jak również elementarne mechanizmy nadzoru nad jej poprawnością (co ma duże znaczenie w warunkach przemysłowych, gdzie liczne zakłócenia mogą powodować błędy transmisji, z czego wynika również ograniczona jej szybkość). Problemem może być zdobycie aplikacji do transmisji szeregowej, jednak często są one dodatkowym wyposażeniem systemów CAM, wiele prostych programów typu freeware można również znaleźć w Internecie. Do ProgMastera w wersji komercyjnej również dołączony jest prosty program transmisyjny, który w niniejszym rozdziale krótko zostanie omówiony.

11.2.1. Standardy kodowania danych

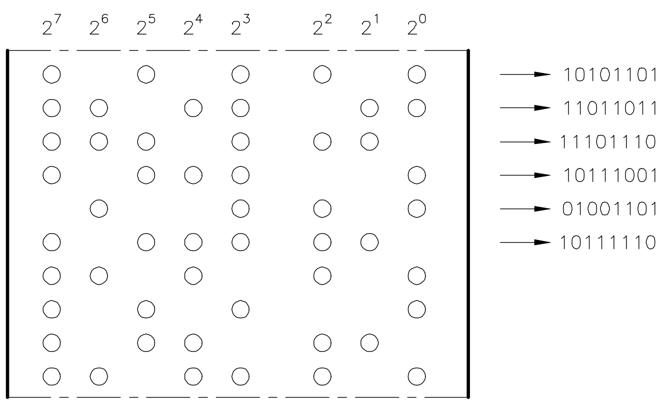
Naturalne dla człowieka znaki alfanumeryczne (litery, cyfry) nie są zrozumiałe dla komputera jako maszyny cyfrowej. Aby komunikacja człowiek-komputer mogła dojść do skutku konieczne jest kodowanie tych znaków do postacji liczbowej (a ściślej binarnej), zrozumiałej dla komputera.

W technice komputerowej najbardziej znanym i szeroko stosowanym jest standard kodowania ASCII (ang. *American Standard Code of Information Interchange*). Początkowo był on kodem 7 bitowym, obecnie stosowana jest jego wersja 8-bitowa, pozwalająca na zakodowanie 256 znaków sterujących i alfanumerycznych. Pierwsze 32 znaki (o kodach $0 \div 31$) są znakami sterującymi (nie posiadają ustalonej reprezentacji graficznej), przeznaczone do celów kontroli transmisji. Najważniejsze z nich przedstawiono w Tabl. 3.

Tabl. 3. Najważniejsze znaki sterujące w kodzie ASCII

Kod dziesiętny	Kod heksadecymalny	Nazwa angielska	Znaczenie
2	02	Start of Text	Początek danych (tekstu)
3	03	End of Text	Koniec danych (tekstu)
4	04	End of Transmission	Koniec transmisji
10	0A	Line Feed	Przejście do następnego wiersza
13	0D	Carriage Return	Przejście do pierwszej kolumny
17	11	Device Control 1	Znak XON
19	13	Device Control 3	Znak XOFF

Tak duża liczba znaków nie jest wymagana w układach sterowania (wystarczą cyfry, znaki alfabetu łacińskiego i kilka dodatkowych znaków – razem ok. 50 znaków). Stąd od szeregu lat układy sterowania korzystają z 7-bitowych systemów kodowych, stanowiących podzbiór (pierwsze 128 znaków) kodu ASCII. Należą do nich kody **ISO** (oparty o normę DIN 66025) oraz **EIA** (ang. *Electronic Industries Association*, zdefiniowany przez normę EIA RS-244), przy czym oba kody wykorzystują do zapisu danych 8 bitów. Jest to konsekwencją stosowania w starszych generacjach układów sterowania mało trwałych nośników programów, np. w postaci papierowej taśmy perforowanej. Brak perforacji na danej pozycji był odczytywany jako logiczne 0, wyperforowany otwór był z kolei odczytywany jako logiczna 1 – Rys. 199.



Rys. 199. Zapis danych na taśmie perforowanej

Zarówno na etapie perforowania, jak i eksploatacji mogło dojść do przekłamania zapisu. Aby temu zapobiec zakodowany znak był uzupełniany o bit (nie)parzystości (bit na pozycji 2^7). jego wartość była tak dobierana, że w kodzie ISO (zwanym kodem parzystym) łączna liczba jedynek w 8-bitowym zapisie znaku była liczbą parzystą, natomiast w kodzie EIA (zwanym też nieparzystym) na odwrót – była liczbą nieparzystą. W Tabl. 4 przedstawiono przykłady kodowania znaków alfanumerycznych w obu omawianych kodach.

Tabl. 4. Przykład kodowania znaków alfanumerycznych w kodzie ISO i EIA

Znak	Kod ASCII	Kod ISO	Kod EIA
%	37	10100101	00100101
1	49	10110001	00110001
9	57	00111001	10111001
B	66	01000010	11000010
G	71	01000111	11000111
M	77	01001101	11001101
X	88	11011000	01011000

Jeżeli w trakcie odczytu znaku w jednym z omawianych kodów wykryto niezgodność liczby znaków 1 ze standardem kodu to oznaczało to błędną postać odczytanego znaku. Jest to zatem najprostszy sposób weryfikacji poprawności transmisji danych (przy założeniu bardzo małego prawdopodobieństwa wystąpienia większej liczby przekłamań w transmisji jednego znaku).

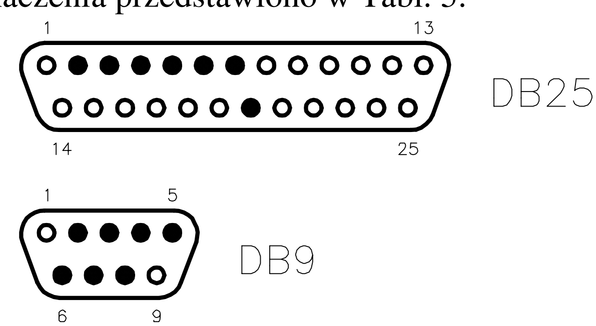
11.2.2. Transmisja szeregowa asynchroniczna

Wymiana danych poprzez łącze szeregowe może odbywać się dwoma metodami:

1. Synchroniczną, kiedy dane przesyłane są w ściśle określonych odstępach czasu, konieczność odbioru danych przez odbiornik jest synchronizowana przez nadajnik dodatkowym sygnałem zegarowym.
2. Asynchroniczną, kiedy dane są przesyłane w przypadkowych momentach czasu, nadajnik musi automatycznie wykryć nadchodzące od nadajnika dane i rozpocząć ich odbieranie.

Z uwagi na większą prostotę funkcjonowania stosowana jest przede wszystkim transmisja asynchroniczna. Aby zrozumieć zasadę jej realizacji należy poznać budowę interfejsu szeregowego.

Złącze RS 232C występuje w postaci 9-pinowej (DB9) lub 25-pinowej (DB25) – Rys. 200 (aktywne końcówki oznaczono kolorem czarnym). Ich oznaczenia symboliczne i opis znaczenia przedstawiono w Tabl. 5.

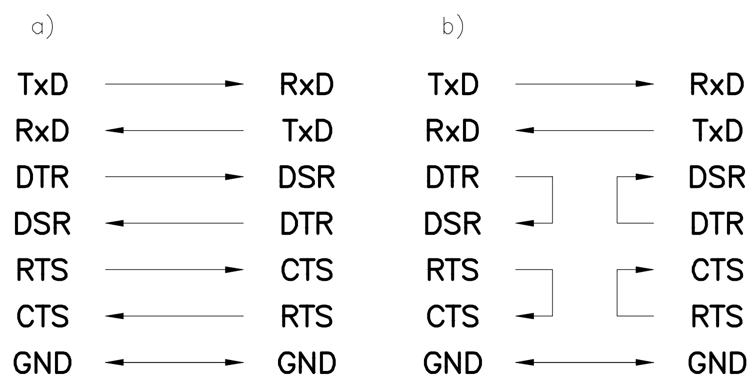


Rys. 200. Gniazda szeregowe DB9 i DB25 (od strony komputera)

Tabl. 5. Opis aktywnych końcówek złącz szeregowych

DB25	DB9	Symbol	Nazwa	Opis	Kierunek
2	3	TxD	*Transmitted Data*	dane wysyłane	wyjście nadajnika
3	2	RxD	*Received Data*	dane odbierane	wejście odbiornika
4	7	RTS	*Request To Send*	gotowość nadajnika do wysyłania danych	wyjście nadajnika
5	8	CTS	*Clear To Send*	gotowość odbiornika do odbierania danych	wejście odbiornika
6	6	DSR	*Data Set Ready*	gotowość odbiornika do nawiązania połączenia	wyjście odbiornika
7	5	GND	*Ground*	masa sygnałowa	
20	4	DTR	*Data Terminal Ready*	gotowość nadajnika do nawiązania połączenia	wyjście nadajnika

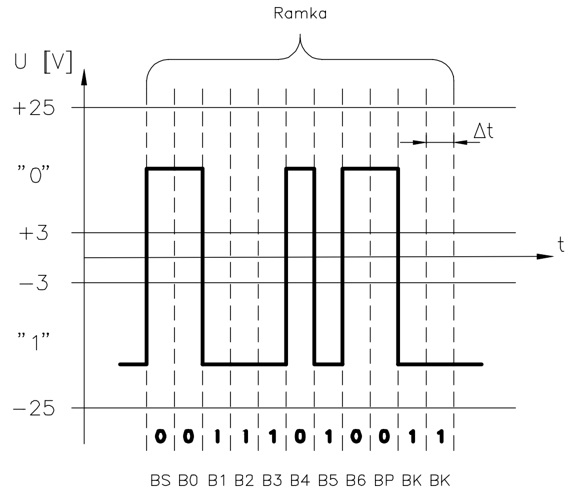
Przy kojarzeniu dwóch urządzeń łączy się odpowiednie końcówki – Rys. 201a. Połączenia tam pokazane stanowią kompletny zestaw. W praktyce często ogranicza się zestaw używanych połączeń, niezbędne minimum do transmisji dwukierunkowej obejmuje połączenia trzech sygnałów: TxD, RxD oraz GND (Rys. 201b).



Rys. 201. Połączenia końcówek dwóch urządzeń transmitujących dane łączem szeregowym w wersji pełnej (a) oraz uproszczonej (b)

Łącze szeregowe RS 232C jest interfejsem cyfrowym. Przez linie łączące wejścia i wyjścia są transmitowane sygnały napięciowe, odpowiadające logicznemu zeru i jedynce. Jedynce logicznej (″1″) odpowiada przedział napięć -3 ÷ -25 V, zaś zeru (″0″) +3 ÷ +25 V. Taki układ napięć jest konsekwencją przyjętego sposobu przesyłu danych. Odbywa się ona z wykorzystaniem tzw. ramki, która zawiera (Rys. 202):

1. bity startu (BS);
2. bity danych (B0, B1, B2, ...);
3. bity parzystości (BP);
4. bity stopu (BK).



Rys. 202. Czasowy przebieg napięć podczas transmisji jednej ramki

W stanie nieaktywnym łącze znajduje się w stanie logicznej jedynki. Rozpoczęcie nadawania sygnalizowane jest **dwoma bitami startu** (BS), których wartość zawsze jest równa 0. Czas, przez jaki są nadawane bity startu służy odbiornikowi do synchronizacji momentu rozpoczęcia odbioru bitów danych (B0, B1, ...). Następnie może być przesłany bit parzystości (BP), który pełni podobną rolę jak bit parzystości w kodach ISO i EIA, przy czym jest on wyznaczany w odniesieniu do wszystkich bitów danych, znajdujących się w ramce (**niezależnie od sposobu kodowania przesyłanego znaku**). Na zakończenie nadajnik wysyła bity stopu (BK), równe wartości 1, które ustawiają linię przesyłu w stan nieaktywny. Teraz może rozpocząć się transmisja kolejnej ramki.

Aby zachować poprawność transmisji danych zarówno nadajnik jak i odbiornik musi stosować identyczny format ramki. Przed przeprowadzeniem transmisji należy zatem dokonać definicji parametrów transmisji. Należą do nich:

1. Liczba bitów danych – dopuszczalne wartości to 5, 6, 7 lub 8;
2. Rodzaj bitu parzystości – dostępne wartości to:
   - ➔ None – brak bitu parzystości;
   - ➔ Odd – łączna liczba jedynek bitów danych i bitu parzystości jest nieparzysta;
   - ➔ Even – łączna liczba jedynek bitów danych i bitu parzystości jest parzysta;
   - ➔ Mark – bit parzystości zawsze równy 1;
   - ➔ Space – bit parzystości zawsze równy 0;
3. Liczba bitów stopu – dopuszczalne wartości to 1, 1.5 (tylko dla 5 bitów danych), 2.
4. Szybkość transmisji – wyrażona w [bit/s], zwanych inaczej [baud] lub po polsku [bod]; typowe wartości to 110, 300,600, 1200, 2400, 4800, 9600 itd.

Ostatni z parametrów związanych jest z czasem transmisji jednego bitu (wartość Δt [s/bit] na Rys. 202). Zarówno po stronie nadajnika jak i odbiornika czas ten musi być identyczny. Dla większej przejrzystości podaje się odwrotność czasu transmisji jednego bitu, wyrażoną w [bit/s]. Należy zauważyć, że efektywna szybkość transmisji (tj. przesyłu bitów danych), jest mniejsza z uwagi na obecność w ramce dodatkowych bitów.

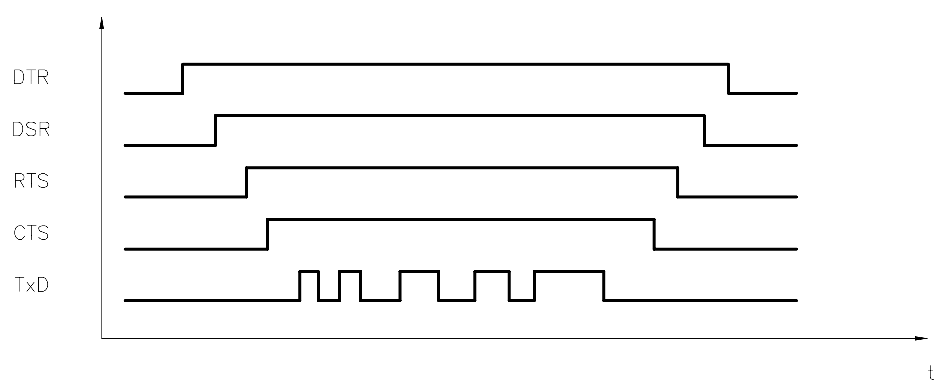
11.2.3. Kontrola przepływu danych

Transmisja asynchroniczna cechuje się tym, iż przepływ danych odbywa się w losowych momentach czasu. Może się zdarzyć, że nadajnik nadaje szybciej niż odbiornik może odbierać (np. w drukarkach z interfejsem szeregowym wydruk jest wolniejszy niż przepływ danych do drukarki) mimo stosowania mechanizmów zapobiegających temu zjawisku – np. buforów. Odbiornik powinien mieć zatem możliwość poinformowania nadajnik o konieczności chwilowego wstrzymania transmisji, np. do czasu opróżnienia bufora wydruku.

Przewidziano dwie możliwości sterowania transmisją danych:

- ➔ Programową – odbiornik w momencie zapełnienia bufora w stopniu bliskim 100% wysyła do nadajnika znak informujący o konieczności chwilowego wstrzymania transmisji – XOFF. Standardowo jest to znak o kodzie dziesiętnym równym 19 (Tabl. 3). Jeżeli zapełnienie bufora odbiornika zbliży się do wartości 0% wysyła on do nadajnika znak pozwalający na ponowne uruchomienie transmisji – XON. Standardowo jest to znak o kodzie dziesiętnym 17 (Tabl. 3). Taki sposób kontroli jest możliwy, jeżeli przesyłane dane nie zawierają znaków sterujących (z przedziału 0 ÷ 31 kodów ASCII) – dotyczy to przesyłu plików tekstowych, a więc również programów sterujących. Dla przesyłu plików binarnych ta metoda nie może być stosowana (znaki sterujące XON i XOFF mogą wystąpić wśród danych). Potocznie kontrolę programową nazywa się protokołem XON-XOFF.
- ➔ Sprzętową – wykorzystuje się dodatkowe linie przesyłowe (kontrola łącza DTR-DTS oraz RTS-CTS). W tym wypadku sterowanie może dotyczyć zarówno transmisji plików tekstowych jak i binarnych. Metoda sprzętowa charakteryzuje się ponadto większą skutecznością, stąd jest stosowana częściej niż programowa.

Dla wyjaśnienia funkcjonowania sprzętowej kontroli transmisji należy zanalizować dokładnie przebieg czasowy sygnałów sterujących – Rys. 203.



Rys. 203. Przebieg sygnałów sterujących nadajnika podczas transmisji

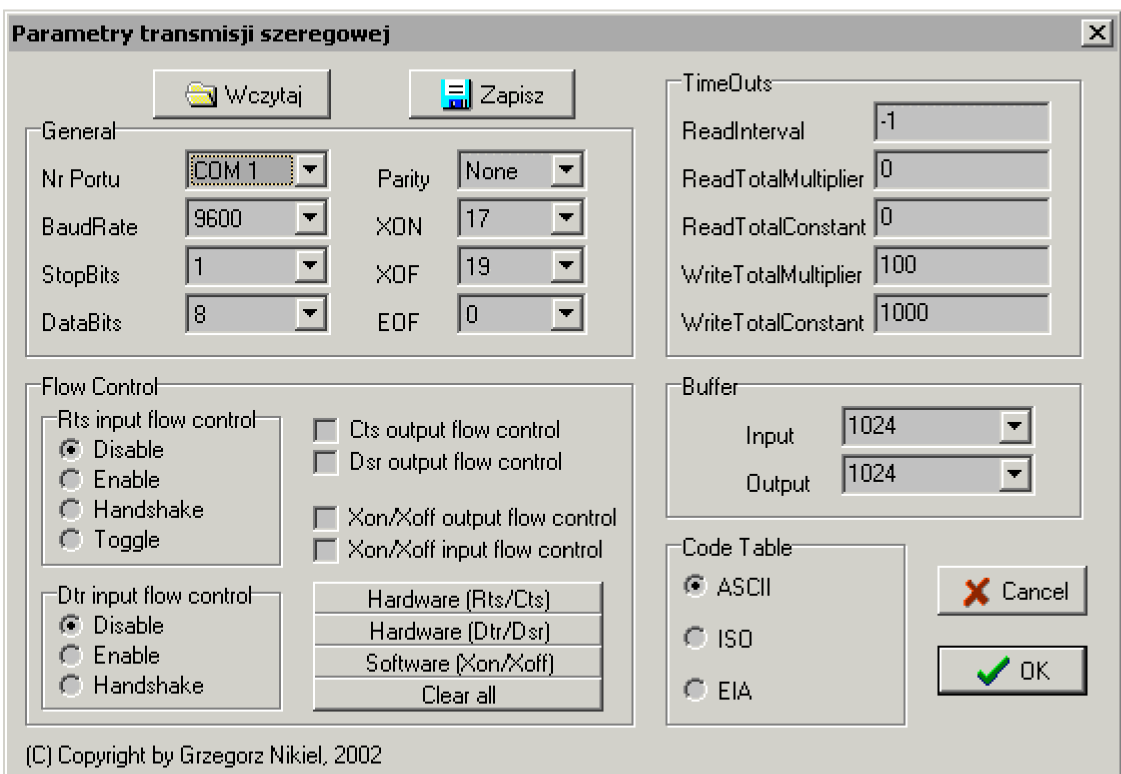
W stanie nieaktywnym sygnały wyjściowe DTR i RTS nadajnika są w stanie niskim. Jeżeli nadajnik zamierza rozpocząć transmisję danych ustanawia stan wysoki na wyjściu DTR, połączonym z wejściem DSR odbiornika (Rys. 201a). Jeżeli odbiornik jest gotowy do nawiązania połączenia na swoim wyjściu DTR (połączonym z wejściem DSR nadajnika) ustawia stan wysoki. Po wykryciu tego stanu nadajnik na wyjściu RTS (połączonym z wejściem CTS odbiornika) ustawia stan wysoki. Jeżeli odbiornik jest gotowy do odbioru danych to na swoim wyjściu RTS (połączonym z wejściem CTS nadajnika) ustawia stan wysoki. Dopiero wykrycie przez nadajnik

tego stanu jest równoznaczne z rozpoczęciem przesyłu danych (wyjście TxD). Po zakończeniu transmisji zarówno nadajnik jak i odbiornik na swoich wyjściach DTR i RTS ponownie ustawiają stan niski. Jeżeli w trakcie transmisji odbiornik z powodów wcześniej opisanych musi chwilowo wstrzymać odbiór danych to na swoim wyjściu DTR lub RTS ustawia stan niski – jest to sygnałem do wstrzymania nadawania przez nadajnik.

Powyżej opisany sposób jest rzadko spotykany. Najczęściej stosuje się mostkowanie wyjść i wejść nadajnika i odbiornika (Rys. 201b) skracając opisaną procedurę, a przede wszystkim zmniejszając liczbę niezbędnych do transmisji przewodów. W praktyce można spotkać się z kontrolą tylko linii DTR-DSR (potocznie protokół DTR-DSR), tylko linii RTS-CTS (potocznie protokół RTS-CTS) lub bez kontroli żadnej z tych linii (Rys. 201b). W tym ostatnim przypadku nadajnik wysyła dane bez sprawdzenia, czy nadajnik w ogóle coś odbiera. Ważne jest zatem, aby **wcześniej włączyć odbieranie danych w odbiorniku, zanim uruchomione zostanie nadawanie przez nadajnik** aby odbiornik mógł odebrać wszystkie wysłane dane. Dodatkowo, w takim przypadku należy zdefiniować znak końca transmisji (standardowo jest to znak o kodzie ASCII równym 4 – Tabl. 3). Po odebraniu tego znaku odbiornik przestaje oczekiwać na kolejne dane, przesyłane z nadajnika.

11.2.4. Przykład programu do transmisji szeregowej

Jako przykład aplikacji służącej do przesyłu danych poprzez łącze szeregowe przedstawiony zostanie program o nazwie RS232, dołączony do ProgMastera. Po uruchomieniu aktywna jest plansza konfiguracji parametrów transmisji (Rys. 204).



Rys. 204. Plansza konfiguracji parametrów transmisji szeregowej

Dostępne na niej opcje to:

Wczytaj – wczytanie danych konfiguracyjnych z pliku tekstowego;

Zapisz – zapisanie aktualnych danych konfiguracyjnych do pliku tekstowego;

General – ogólne nastawy pracy łącza szeregowego, w tym:

- ***Nr portu*** – symbol portu szeregowego, przez który będą transmitowane dane;
- ***BaudRate*** – szybkość transmisji [baud];
- ***StopBits*** – liczba bitów stopu (BK);
- ***DataBits*** – liczba bitów danych (B0, B1, ...);
- ***Parity*** – określenie czy występuje bit parzystości (BP) oraz rodzaj kontroli parzystości;
- ***XON*** – kod dziesiętny znaku XON;
- ***XOF*** – kod znaku XOFF;
- ***EOF*** – kod znaku końca transmisji.

TimeOuts – grupa parametrów czasowych związanych z pracą łącza, tu nie będą szczegółowo omawiane, należy pozostawić wartości domyślne.

Buffer – definicja wielkości buforów w [B]:

- ***Input*** – wyjściowego (przy wysyłaniu danych);
- ***Output*** – wejściowego (przy pobieraniu danych).

Code Table – sposób kodowania przesyłanych znaków (***ASCII***, ***ISO***, ***EIA***).

Flow Control – parametry kontroli transmisji, w tym:

- ***Rts input flow control*** – kontrola stanu sygnału RTS;
- ***Dtr input flow control*** – kontrola stanu sygnału DTR;
- ***Cts output flow control*** – kontrola stanu sygnału CTS;
- ***Dsr output flow control*** – kontrola stanu sygnału DSR;

  Powyższe cztery sygnały są brane pod uwagę przy sprzętowej kontroli przepływu danych. Aby ułatwić użytkownikowi dobranie odpowiednich ustawień dodatkowo umieszczono dwa przyciski:

- ***Hardware (Rts/Cts)*** – ustawienie danych kontroli przepływu dla protokołu RTS-CTS;
- ***Hardware (Dtr/Dsr)*** – ustawienie danych kontroli przepływu dla protokołu DTR-DSR;

  których naciśnięcie spowoduje automatyczne ustawienie parametrów kontroli transmisji.

- ***Xon/Xoff output flow control*** – kontrola programowa danych wysyłanych;
- ***Xon/Xoff input flow control*** – kontrola programowa danych odbieranych.

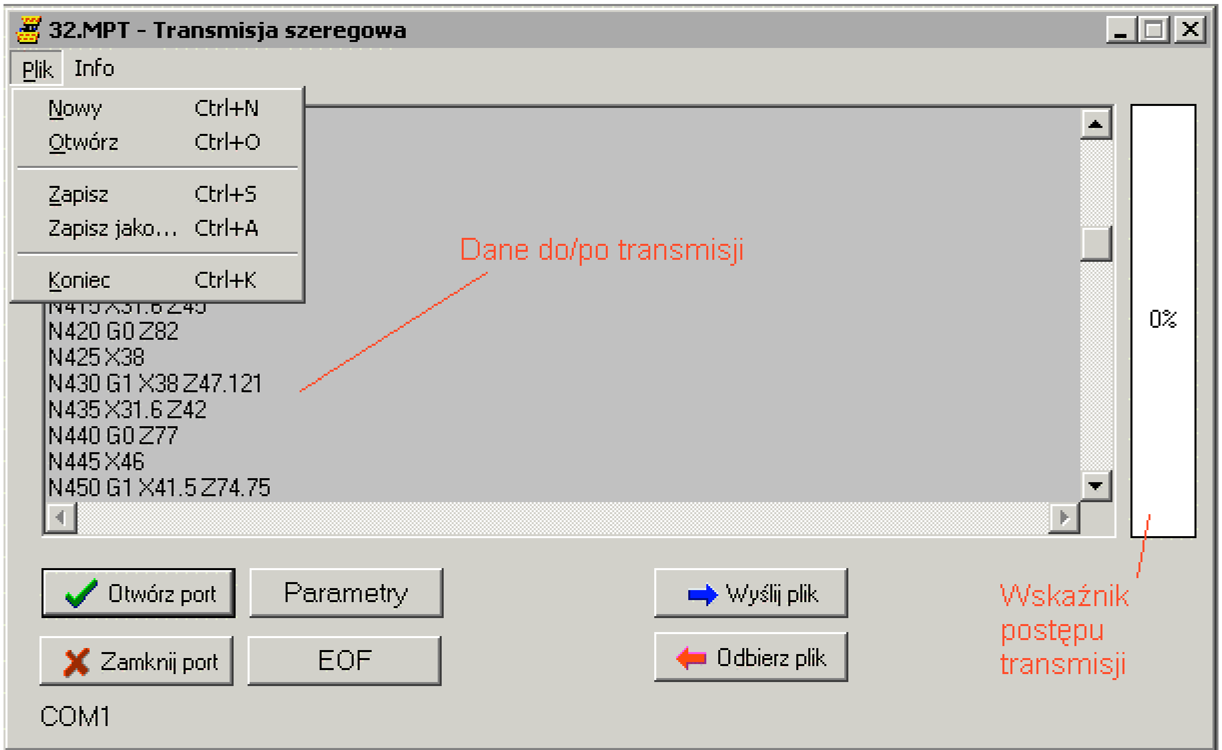
  Podobnie jak dla kontroli sprzętowej umieszczono specjalny przycisk:

- ***Software (Xon/Xoff) –*** ustawienie kontroli programowej transmisji;

  którego naciśnięcie spowoduje automatyczne ustawienie parametrów kontroli transmisji.

- ***Clear all –*** naciśnięcie tego przycisku kasuje wszystkie poczynione wcześniej ustawienia dotyczące kontroli transmisji – brak wszelkiej kontroli.

Po poprawnym zdefiniowaniu parametrów transmisji przyciskiem ***OK*** przechodzimy do planszy transmisji danych (Rys. 205).



Rys. 205. Plansza główna programu do transmisji szeregowej asynchronicznej

Dostępne opcje to:
Otwórz port – jeżeli wszystkie ustawione parametry transmisji są poprawne to wskazany na dole (pod przyciskiem ***Zamknij port***) jest otwarty i gotowy do transmisji. Jeżeli jakiś parametr był niewłaściwy (np. brak wskazanego portu) to program wyświetli komunikat błędu i nie uaktywni portu komunikacyjnego.
Zamknij port – powoduje zamknięcie otwartego portu komunikacyjnego co umożliwia np. zmianę parametrów transmisji.
Parametry – przejście do planszy konfiguracji parametrów transmisji (Rys. 204).
EOF – wysłanie znaku końca transmisji (w sytuacji gdy np. doszło do przerwania wysyłania danych).
Wyślij plik – rozpoczęcie wysyłania danych zgodnie z ustawionymi parametrami transmisji. Transmitowane są dane zawarte w okienku *Dane do/po transmisji* (pobrane z pliku dyskowego). Na *Wskaźniku postępu transmisji* podawane jest procentowo zaawansowanie transmisji.
Odbierz plik – rozpoczęcie odbioru danych zgodnie z ustawionymi parametrami transmisji. Odebrane dane są umieszczane w okienku *Dane do/po transmisji* (później można je zapisać do pliku dyskowego). Podgląd danych podczas transmisji pozwala zorientować się, czy przebiega ona prawidłowo. Najczęstszym błędem transmisji są różne parametry transmisji po stronie nadajnika i odbiornika.
Plik – grupa funkcji zarządzająca danymi wysyłanymi/odbieranymi:

> ***Nowy*** – kasuje wszystkie dane w okienku *Dane do/po transmisji*;
> ***Otwórz*** – otwiera plik dyskowy i wczytuje jego zawartość do okienka *Dane do/po transmisji*;
> ***Zapisz*** – zapisuje dane z okienka *Dane do/po transmisji* w pliku dyskowym o bieżącej nazwie (podawane w nagłówku programu);

Zapisz jako – zapisuje dane z okienka *Dane do/po transmisji* w pliku dyskowym o nazwie podanej przez użytkownika;

Koniec – zakończenie pracy z programem.

Z uwagi na przeznaczenie aplikacji do transmisji programów sterujących, które są plikami tekstowymi, nie są obsługiwane pliki binarne.